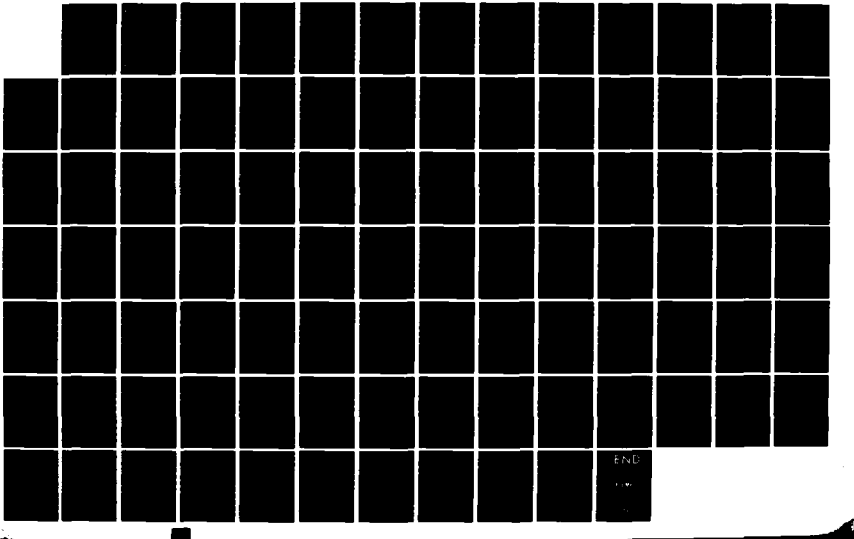
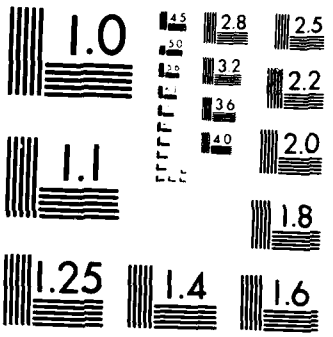
AD-A151 896 INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DET..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D E OYLER 1/1

UNCLASSIFIED DEC 84 AFIT/GEP/PH/84D-7 F/G 12/1 NL





INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DETERMINATION OF DIFFERENTIAL EQUATIONS

THESIS

Dean E. Oyler
First Lieutenant, USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DETERMINATION OF DIFFERENTIAL EQUATIONS

THESIS

Dean E. Oyler
First Lieutenant, USAF

AFIT/GEP/PH/84D-7

INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DETERMINATION OF DIFFERENTIAL EQUATIONS

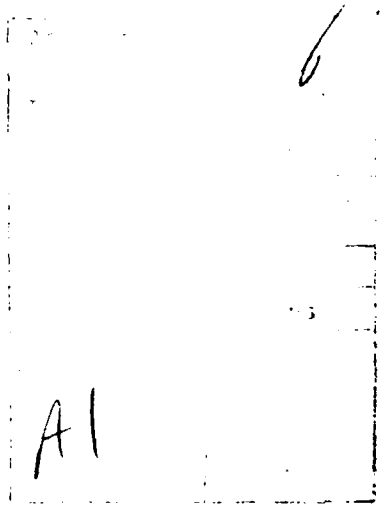
THESIS

Presented to the Faculty of the School of Engineering of
the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Dean E. Oyler, B.A., B.S.
1/Lt, USAF

December 1984



Preface

This report is the result of a twelve week study on the feasibility of using the method of weighted residuals to determine approximations to the discrete Green's function or an analog to it. Included in the report are derivations of the methods of Galerkin, collocation and finite differences, for the one- and two-dimensional Poisson's equation. The analytical solutions for various inhomogeneity terms are also included. The problem of ill-conditioned matricies which arose in two cases is discussed in Appendix A. All of the primary goals of the study were met or explained.

During the twelve week period, I learned a great deal about the theory of Green's functions, numerical methods, matricies and the problems that can come about from solving matrix equations.

I would like to acknowledge Dr. Kaplan for his support, and seemingly never-ending list of reference sources. His suggestions always opened up new avenues for research, and taught me more about the subjects than I really wanted to know.

I must also thank my wife, Roxann, for her understanding during this stressful period, and for being willing to share the computer with me despite needing it for work on her own thesis.

Dean E. Oyler

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this study was to determine the feasibility of using the method of weighted residuals to obtain approximations to the discrete Green's function, or analogs to it. The weighted residual methods of Galerkin and collocation, as well as the finite difference method were programed on a Kaypro II micro-computer in Microsoft Basic. These programs were used to generate approximations to the one- and two-dimensional Poisson's equation. The two-dimensional case was restricted to the geometry of a unit square. Various inhomogeneity terms were used to obtain approximate solutions to the discrete Green's functions or their analogs. The results were compared with the analytical values at various interior nodal points on the mesh. The average percent error for the approximations were reported for each case as the number of interior nodal points was increased. The areas of consideration were the rate of convergence to the analytical solution, the amount of time it took to run each program, and the accuracy of the approximate solutions. The results of this study indicate that the Green's functions or analogs obtained are valid approximations to the discrete Green's functions. The method of weighted residuals proved to be very sensitive to the choice of basis functions, resulting in ill-conditioned matricies in some instances.

INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DETERMINATION OF DIFFERENTIAL EQUATIONS

I. Introduction

Background

The final step in the mathematical treatment of many problems in physics and engineering is often finding the solution to a boundary-value problem. The standard differential equations most often encountered in mathematical physics include -

LaPlace's equation:

$$\nabla^2\phi = 0 \tag{1}$$

Poisson's equation:

$$\nabla^2\phi = -\rho \tag{2}$$

the wave equation:

$$\nabla^2\psi - 1/c^2[\partial^2\psi/\partial t^2] = 0 \tag{3}$$

and the Helmholtz equation:

$$(\nabla^2 + k^2)\psi = f \tag{4}$$

Frequently, the solutions to these equations can be represented in terms of a Green's function. There are several advantages in the use of Green's functions as solutions to these boundary-value problems. One advantage is that it enables a differential equation with suitable

boundary conditions to be solved by an ordinary integral. Another advantage is that once the Green's function for a particular differential operator and geometry has been found, it can be utilized for all other problems involving the same differential operator and geometry, but with different expressions for the inhomogeneity or source term. If these Green's functions for different differential operators and geometries could be tabulated, they could be used to solve boundary-value problems quite easily, in a manner analogous to the use of a table of integrals.

Although Green's functions have been obtained analytically for certain standard geometries (planes, rectangles, spheres, cylinders), for the usual equations of mathematical physics (Eqs(1-4)), a difficulty arises in finding the Green's functions for mixed or irregular geometries. In these situations, one must employ the use of numerical methods techniques.

Previous thesis research and publications have successfully solved the problem of numerically approximating Green's functions by means of finite difference algorithms (1,3,6). In this method, one uses approximations of derivatives (usually a truncated Taylor's series) to convert the boundary-value problem into a large series of simultaneous algebraic equations, which can than be solved with relative ease using matrix methods on a digital computer.

This thesis will investigate the use of a different numerical technique, the method of weighted residuals, to solve the necessary differential equations. In this technique, the unknown solution is expressed as a series of functions which can be manipulated to once again reduce the problem to solving a series of simultaneous algebraic

equations. The inverse of the coefficient matrix of these equations is analogous to the discrete Green's function for the differential operator.

Objective

This research effort will compare the discrete Green's functions, obtained by both the method of weighted residuals and the method of finite differences, for both the one- and two-dimensional case, with respect to accuracy and feasibility for digital computation.

The current thesis problem is a follow-up to a previous M.S. thesis (1) that reported conflicting results for the one- and two-dimensional cases, concerning which of the three methods was best. This study will attempt to verify or refute the conflict between the two cases by recreating parts of the previous thesis using a different computer code, and different matrix solving routines.

Scope

This study will only consider the problem of the one- and two-dimensional Poisson's equation, with Dirichlet boundary conditions. The solutions for both the one- and two-dimensional Green's functions will be compared using the finite difference method and the method of weighted residuals.

Approach

The initial approach to this study will be to develop computer programs which use the method of finite differences and two of the

methods of weighted residuals (Galerkin's method, and collocation) to obtain approximations for both the discrete Green's function, or its analog, and the solution for the one-dimensional Poisson's equation. Homogeneous Dirichlet boundary conditions will be assumed for all cases.

Once the initial programs have been developed, they will then be modified to handle the two-dimensional cases.

The usefulness of the Green's functions obtained in the previous steps will then be analyzed by varying the inhomogeneity term for the Poisson's equation. Areas of consideration will include the number of calculations required, computer run time for each method, the convergence rate to the correct solution, the overall accuracy of the approximations as compared to the analytical solution, and how the results compare to the earlier study (1).

Finally, the feasibility and possible directions for continued research into these approximation methods will be explored.

II. POISSON'S EQUATION IN ONE-DIMENSION

The initial problem examined in this study is the one-dimensional Poisson's equation. The general form of the problem can be expressed as

$$L\,u(x) = g(x) \tag{5}$$

where L is the linear differential operator, d^2/dx^2, g(x), the inhomogeneity term, is the source or excitation (a known function), and u(x) is the field or response (the unknown function to be determined) (7:1-2).

Associated with the problem in this study are the homogeneous Dirichlet boundary conditions

$$u(0) = 0 \tag{6-a}$$

$$u(1) = 0 \tag{6-b}$$

Analytical Solution

The general solution to Eq(5), with an inhomogeneity term of the form

$$g(x) = Ax^2 + Bx + C \tag{7}$$

can be found by direct integration to be

$$u(x) = Ax^4/12 + Bx^3/6 + Cx^2/2 + Dx + E \tag{8}$$

By applying the boundary conditions (Eq (6)), Eq (8) becomes

$$u(x) = Ax^4/12 + Bx^3/6 + Cx^2/2 - (A+2B+6C)x/12 \tag{9}$$

which is the analytical solution to the one-dimensional Poisson's equation.

Numerical Approximations

All of the numerical approximations in this study make use of a technique in which a mesh is superimposed over the region of interest of the problem. For the one-dimensional case, this merely involves subdividing the region by N equally spaced interior nodes. The numerical method is then applied at these nodes resulting in a set of N simultaneous algebraic equations, which can be solved to give the approximate solution to the problem.

The accuracy of the approximation depends upon the number of interior nodal points used. A fine mesh with many nodal points will generally result in a more accurate solution to the problem. A sample mesh for the one-dimensional case is shown below in Figure 1

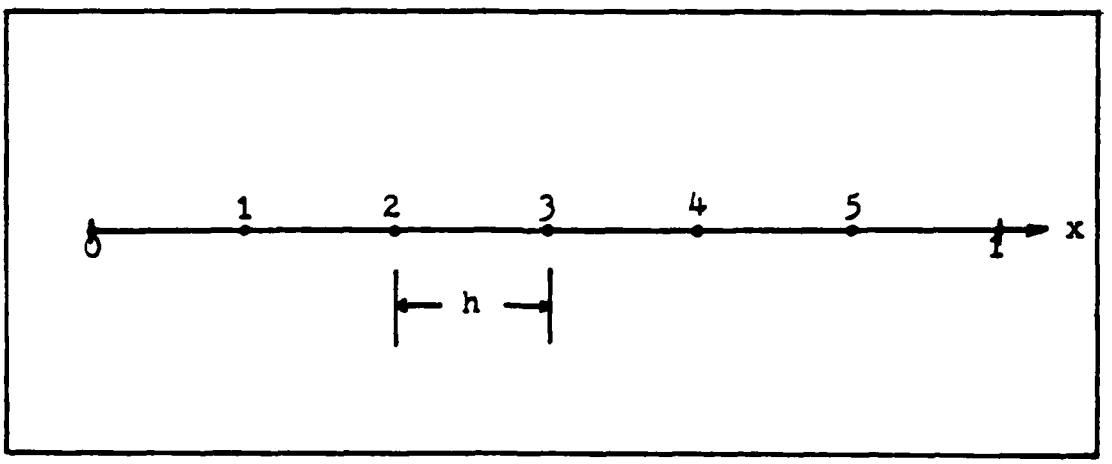


Figure 1. 1-D Mesh With 5 Interior Nodes

The mesh in Figure 1 has five interior nodes, with a step size h between each of the nodes equal to $1/(N+1)$, to ensure that they are equally spaced.

Finite Difference Method. The method of finite differences makes use of a truncated Taylor's series to approximate the solution to the problem. According to Taylor's theorem (14:6), when a function u, and its derivatives are single-valued, finite and continuous functions of x, then

$$u(x+h) = u(x) + hu'(x) + h^2u''(x)/2 + h^3u'''(x)/6 + \ldots \quad (10)$$

and

$$u(x-h) = u(x) - hu'(x) + h^2u''(x)/2 - h^3u'''(x)/6 + \ldots \quad (11)$$

Adding these two expressions yields

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + \text{higher order terms} \quad (12)$$

Neglecting the higher order terms and solving for $u''(x)$

$$u''(x) = d^2u/dx^2 = (u(x+h) - 2u(x) + u(x-h))/h^2 \quad (13)$$

This equation allows an approximation to Eq (5) to be made so that

$$(u(x+h) - 2u(x) + u(x-h))/h^2 = g(x) \quad (14)$$

By applying this approximation to each nodal point in the mesh, the equation becomes a series of N simultaneous algebraic equations which can then be solved by matrix techniques. These equations can be written

in matrix notation as

$$\mathbf{C}\,\mathbf{u} = \mathbf{g} \tag{15}$$

In all cases when vector notation is used, a boldfaced capital letter indicates a rectangular matrix, and a boldfaced lower case letter indicates a column vector.

The coefficient matrix, C, formed from these equations is known as a tri-diagonal matrix. A tri-diagonal matrix has non-zero values only along the main diagonal, and the adjacent diagonals both above and below the main diagonal. As an example, the tri-diagonal matrix equation for five interior nodes is shown in Figure 2.

$$\begin{bmatrix} -32 & 16 & 0 & 0 & 0 \\ 16 & -32 & 16 & 0 & 0 \\ 0 & 16 & -32 & 16 & 0 \\ 0 & 0 & 16 & -32 & 16 \\ 0 & 0 & 0 & 16 & -32 \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \\ u(x_4) \\ u(x_5) \end{bmatrix} = \begin{bmatrix} g(x_1) \\ g(x_2) \\ g(x_3) \\ g(x_3) \\ g(x_5) \end{bmatrix}$$

Figure 2. Tri-diagonal F.D. Matrix Equation for 5 Nodes

Method of Weighted Residuals. In the various methods of weighted residuals, the unknown function u(x) from Eq(5) is expanded in a series of functions, $b_1(x)$, $b_2(x)$, $b_3(x)$, . . . in the domain of L, as

$$u(x) = \sum_n a_n b_n(x) \tag{16}$$

where the a_n are constants, and the $b_n(x)$ are expansion or basis functions (7:5-6). The basis functions are chosen so as to match the boundary conditions of the problem. For the boundary conditions of

Eq(6), the basis functions were chosen to be a power series (7:7) of the form

$$b_n(x) = x - x^{n+1} \qquad n = 1,2,3, \ldots \tag{17}$$

By substituting Eq(16) into Eq(5) and due to the linearity of L, the equation can be rewritten as

$$\sum_n a_n L\, b_n(x) = g(x) \tag{18}$$

Now, a set of weighting or testing functions w_1, w_2, w_3,. . . is defined in the range of L (7:6), and the inner product of Eq(13) and the weighting functions w_m is taken so that

$$a_n \langle w_m, L\, b_n(x)\rangle = \langle w_m, g(x)\rangle \qquad m = 1,2,3, \ldots \tag{19}$$

or in the more condensed matrix notation

$$\mathbf{C}\,\mathbf{a} = \mathbf{g} \tag{20}$$

where C is the square coefficient matrix

$$\begin{bmatrix} \langle w_1, L\, b_1(x)\rangle & \langle w_1, L\, b_2(x)\rangle & \ldots \\ \langle w_2, L\, b_1(x)\rangle & \langle w_2, L\, b_2(x)\rangle & \ldots \\ \ldots & \ldots & \ldots \end{bmatrix} \tag{21}$$

a is the column vector

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix} \tag{22}$$

and **g** is the column vector

$$\begin{bmatrix} \langle w_1, g_1(x) \rangle \\ \langle w_2, g_2(x) \rangle \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \tag{23}$$

The column vector of constants, **a**, can be determined by

$$\mathbf{a} = \mathbf{C}^{-1}\,\mathbf{g} \tag{24}$$

Once **a** has been determined, the approximate solution can then be written as

$$u(x) = \mathbf{b}^{*}\,\mathbf{a} \tag{25}$$

for a method valid over the entire region, or as

$$\mathbf{u} = \mathbf{B}\,\mathbf{a} \tag{26}$$

for a method valid only at discrete points, where $\mathbf{b}^{*}$ is the transpose vector

$$[b_1, b_2, b_3, \ldots, b_N] \tag{27}$$

$$\mathbf{u} = \begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{bmatrix} \tag{28}$$

and

$$\mathbf{B} = \begin{bmatrix} b_1(x_1), & b_2(x_1), & \ldots, & b_N(x_1) \\ b_1(x_2), & b_2(x_2), & \ldots, & b_N(x_2) \\ \cdots & \cdots & \cdots & \cdots \\ b_1(x_N), & b_2(x_N), & \ldots, & b_N(x_N) \end{bmatrix} . \tag{29}$$

The particular choice made for the weighting functions w_m determines which of the methods of weighted residuals is being used.

Galerkin's Method. The choice of letting the weighting function $w_m = b_n(x)$ is known as Galerkin's method. For $w_m = x - x^{m+1}$, the values for the coefficient matrix, C can be found by taking the inner product of w_m and L $b_n(x)$. The result is

$$C_{mn} = \langle w_m, L\, b_n(x) \rangle = \int_0^1 (x - x^{m+1})\, d^2/dx^2(x - x^{n+1})\, dx \qquad (30)$$

Similarly, g_m is found to be

$$g_m = \langle w_m, g(x) \rangle = \int_0^1 (x - x^{m+1})\, d^2/dx^2(Ax^2 + Bx + C)\, dx \qquad (31)$$

There are no limitations on the value of x, therefore the results obtained by Galerkin's method are valid over the entire region, and not just at the nodal points of the mesh.

Collocation Method. This procedure, also known as point-matching (7:10), makes use of the Dirac delta function as the weighting function. The inner product of w_m and L $b_n(x)$ is then

$$C_{mn} = \langle w_m, L\, b_n(x) \rangle = \int_0^1 \delta(x - x_m)\, d^2/dx^2(x - x^{n+1})\, dx \qquad (32)$$

where x_m are the points equispaced in the interval $0 < x < 1$, ($x_m = m/(N+1)$, $m=1,2,3, \ldots N$) corresponding to the nodal points. And g_m is given by

$$g_m = \langle w_m, g(x) \rangle = \int_0^1 \delta(x - x_m)\, d^2/dx^2(Ax^2 + Bx + C)\, dx \qquad (33)$$

The collocation method limits the value of x to the values at the

nodal points, therefore, the method of collocation approximations are only valid at these nodal points.

Green's Functions and Analogs

This section will show how the various approximation techniques utilized in this study may be related to the Green's function or its analog.

Finite Difference Method. The relationship for the Green's function for the differential operator L of Eq(5) has been defined (15:6-7) to be

$$d^2G(x|x_o)/dx^2 = \delta(x-x_o) \tag{34}$$

where $G(x|x_o)$ is the Green's function for Eq(5), x is the field point, and x_o is the source point. The associated Dirichlet boundary conditions for Eq(34) are

$$G(0|x_o) = 0 \tag{35-a}$$

$$G(1|x_o) = 0 \tag{35-b}$$

When applied to a mesh with step size h_N, Eq(34) takes the form of the discrete Green's function (5:314-315)

$$d^2G_N(x|x_o)/dx^2 = \delta(x-x_o)/h_N \tag{36}$$

where N is once again the number of interior nodal points on the mesh.

The derivative term can then be treated in the same manner as was done earlier for the finite difference method, and be replaced by a central difference equation

$$d^2G_N(x|x_o)/dx^2$$

$$= (G_N(x+h_N|x_o) - 2G_N(x|x_o) + G_N(x-h_N|x_o))/h^2 \qquad (37)$$

Substituting this equation into Eq(36) and multiplying through by h^2, the expression becomes

$$G_N(x+h_N|x_o) - 2G_N(x|x_o) + G_N(x-h_N|x_o) = h_N\,\delta(x-x_o) \qquad (38)$$

Applying Eq(38) along with the associated boundary conditions to each of the nodal points yields a series of N^2 simultaneous algebraic equations which can be expressed in matrix notation as

$$C\,G_N = h_N\,I_N \qquad (39)$$

where C is once again the coefficient matrix, G is the discrete Green's function matrix, and I_N is the identity matrix of order N. The coefficient matrix of the finite difference method of Eq(15) is equivalent to the inverse of the numerical Green's function matrix multiplied by h. The finite difference equations and the Green's function equations yield identical approximations to the solutions.

Method of Weighted Residuals. For the one-dimensional Poisson's equation (Eq(5)), and its associated boundary conditions (Eq(6)), the Green's function for the problem can be determined analytically (15:1-12). The solution to Eq(5) with its various inhomogeneity terms can be found by calculating the integral

$$u(x) = \int_o^l G(x|x_o)\,g(x)\,dx_o \qquad (40)$$

where $G(x|x_o)$ is the Green's function for Eq(5) and its associated boundary conditions, x is the field point, and x_o is the source point.

Eq(40) can be written in matrix notation, for the discrete Green's function on a mesh of step size h_N as

$$\tilde{\mathbf{u}}_N = h\, \mathbf{G}_N\, \mathbf{g}_N \tag{41}$$

where $\tilde{\mathbf{u}}$ is the column vector of solutions at discrete points on the mesh for the given inhomogeneity. The tilde has been placed over the left-hand side of the equation to stress the fact that the discrete Green's function solution may not necessarily be equal to the solution obtained by the method of weighted residuals for u.

If Eq(24) is substituted into Eqs(25) and (26), they can be written as

$$u_N = \mathbf{b}^* \mathbf{C}^{-1}\, \mathbf{g} \tag{42}$$

and

$$u_N = \mathbf{B}\, \mathbf{C}^{-1}\, \mathbf{g} \tag{43}$$

These two equations for the weighted residual approximations do not contain the factor for the step size, h_N, as does Eq(41) This is due to the fact that the weighted residual approximations involve a summation over N terms, while the discrete Green's function solution was developed by approximating an integral equation (Eq(40)), where the step size, h_N, corresponds to the dx term. The analog to the discrete Green's function in Eqs(42) and (43) can be defined to be

$$\bar{\mathbf{G}}_N^* = \mathbf{b}^* \mathbf{C}^{-1} \tag{44}$$

for a method valid over the entire region of interest, such as Galerkin's method (the bar over the G is used to indicate a column vector in this case to avoid confusion with the inhomogeneity term), and

$$G_N^* = B\, C^{-1} \tag{45}$$

for a method valid only at discrete points, such as the collocation method. The superscript asterisk indicates that they are analogs to the discrete Green's function. They are considered analogs since the elements that make up the inhomogeneity vector, g, in Eq(23) may not necessarily be equal to the inhomogeneity term g(x) because of the weighting factor.

Computer Analysis

All of the numerical approximation routines were developed on a Kaypro II microcomputer in Microsoft Basic, using double-precision arithmetic. The program listings are included in Appendix C.

Inhomogeneity Terms and Exact Solutions. The same four functions chosen by Clapp (1) for the excitation terms were also used in this study so that comparisons could be made with the results obtained in that earlier study. These four functions were

$$1)\quad L\,u(x) = 10 \tag{46-a}$$

$$2)\quad L\,u(x) = x^2 \tag{46-b}$$

$$3)\quad L\,u(x) = x^2 + 1 \tag{46-c}$$

$$4)\quad L\,u(x) = x^2 + x + 1 \tag{46-d}$$

with the associated boundary conditions

$$u(0) = 0 \tag{47-a}$$

$$u(1) = 0 \tag{47-b}$$

The number of interior nodes were carefully chosen so that the solution comparisons could all be made at the same nodal points, x=1/3 and x=2/3.

The analytical solutions to the problem set were found by direct integration to be

$$1)\quad u(x) = 5x^2 - 5x \tag{48-a}$$

$$2)\quad u(x) = x^4/12 - x/12 \tag{48-b}$$

$$3)\quad u(x) = x^4/12 - x^2/2 - 7x/12 \tag{48-c}$$

$$4)\quad u(x) = x^4/12 + x^3/3 + x^2/2 - 3x/4 \tag{48-d}$$

The exact solutions at the comparison points are listed in Table 1

TABLE 1

Exact Solutions to Eq(48)

Problem #	x = 1/3	x = 2/3
1	-1.111111	-1.111111
2	-.026749	-.039095
3	-.137860	-.150260
4	-.187243	-.211934

Average Error. The average percent error was the criterion by which the correctness of the approximations were measured. For the one-dimensional case, the average percent error was defined to be

$$\langle E_N \rangle = \left\{ \frac{|u_N(1/3) - u(1/3)|}{u(1/3)} + \frac{|u_N(2/3) - u(2/3)|}{u(2/3)} \right\} \cdot \frac{100}{2} \tag{49}$$

where u_N is the approximation at a specific point for a given number of nodes, and u is the exact solution.

Comparison of Approximations to Exact Solution. The plots of the average percent error vs. number of interior nodal points are shown in Figures 3-6 for each of the four equations in the problem set. The actual values of the approximations are included in Appendix B.

Except in two instances, both of the methods of weighted residuals (Galerkin and collocation) yielded approximations to the exact solution that were orders of magnitude better than the method of finite differences. In addition they were able to achieve these good approximations using a relatively low number of interior nodal points, whereas the finite difference approximations converged to the exact solution more slowly. The only deviations from this trend were in the case of problem 1 (Eq(46-a)) when all methods did equally well on the average, and in problem 4 (Eq(46-d)), when approximations in the Galerkin routine began to diverge rapidly from the correct solution for eight or more interior nodes.

Computer Run Times. Each program was timed by hand using an electronic stop-watch to obtain values for the computer run time. The times varied by only one or two seconds for the various inhomogeneity terms, therefore the times reported are averages for each method. The plot of program run times vs number of interior nodes is shown in Figure 7.

In all cases, the method of finite differences took the least amount of time to run. This method not only has fewer intermediate calculations than the other methods, but in addition, the coefficient matrix was tri-diagonal; this allowed the use of an extremely efficient routine expressly written to solve tri-diagonal matricies (2:122).

Both methods of weighted residuals took considerably longer to arrive at a solution (in some cases, 10 to 30 times longer), for a given

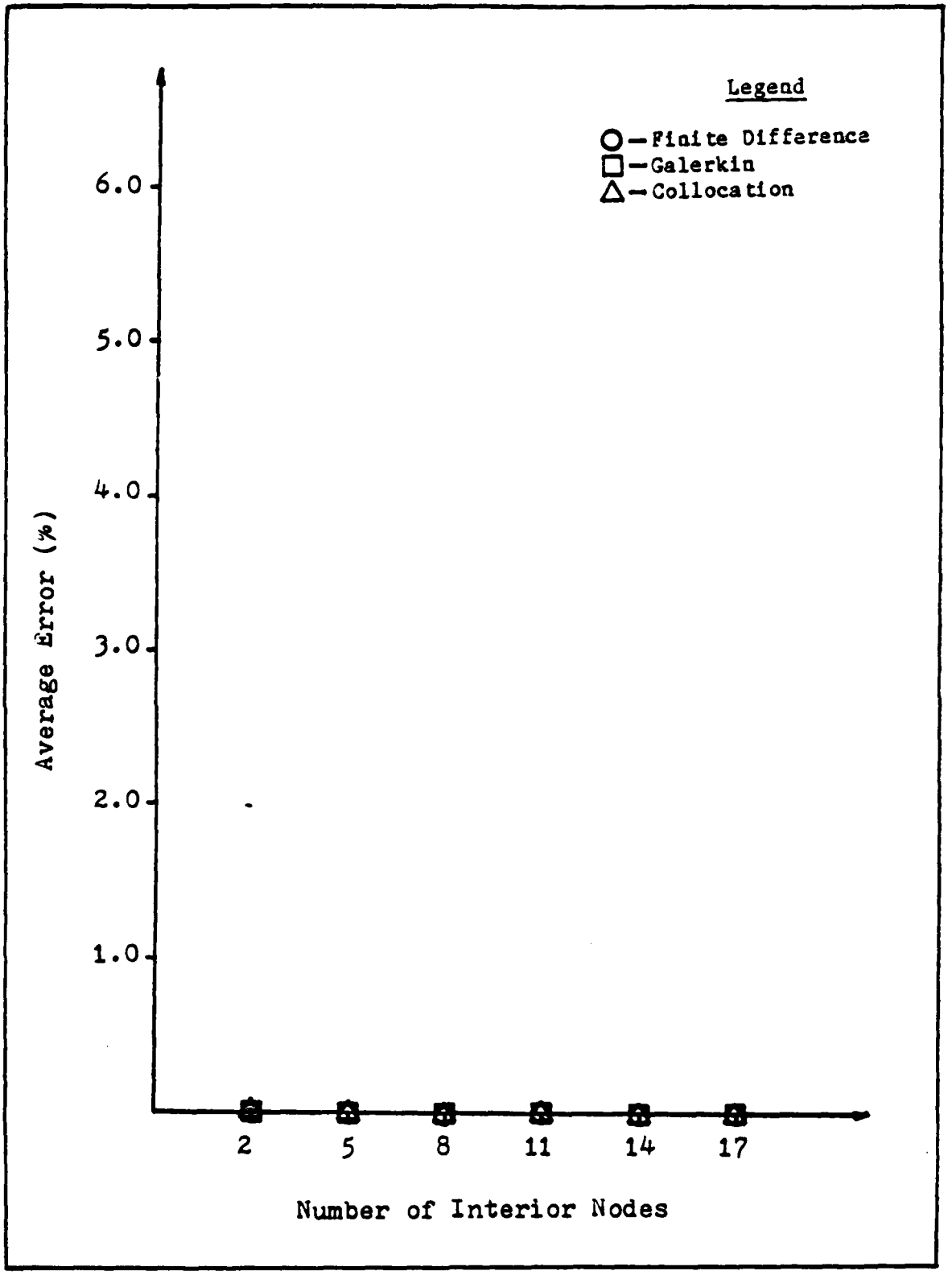


Figure 3. Average Error vs Number of Interior Nodes for g(x) = 10

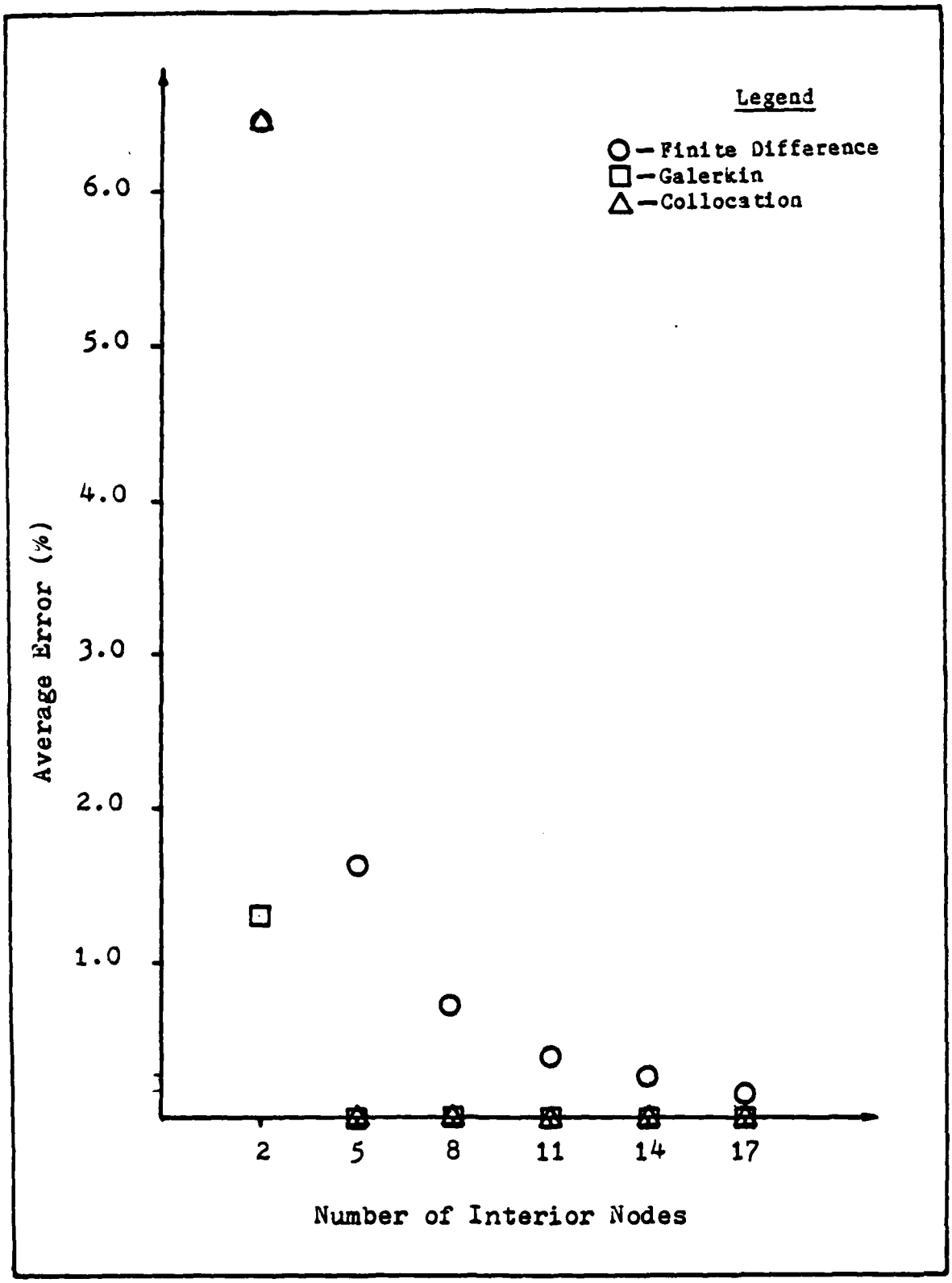


Figure 4. Average Error vs Number of Interior Nodes for $g(x) = x^2$

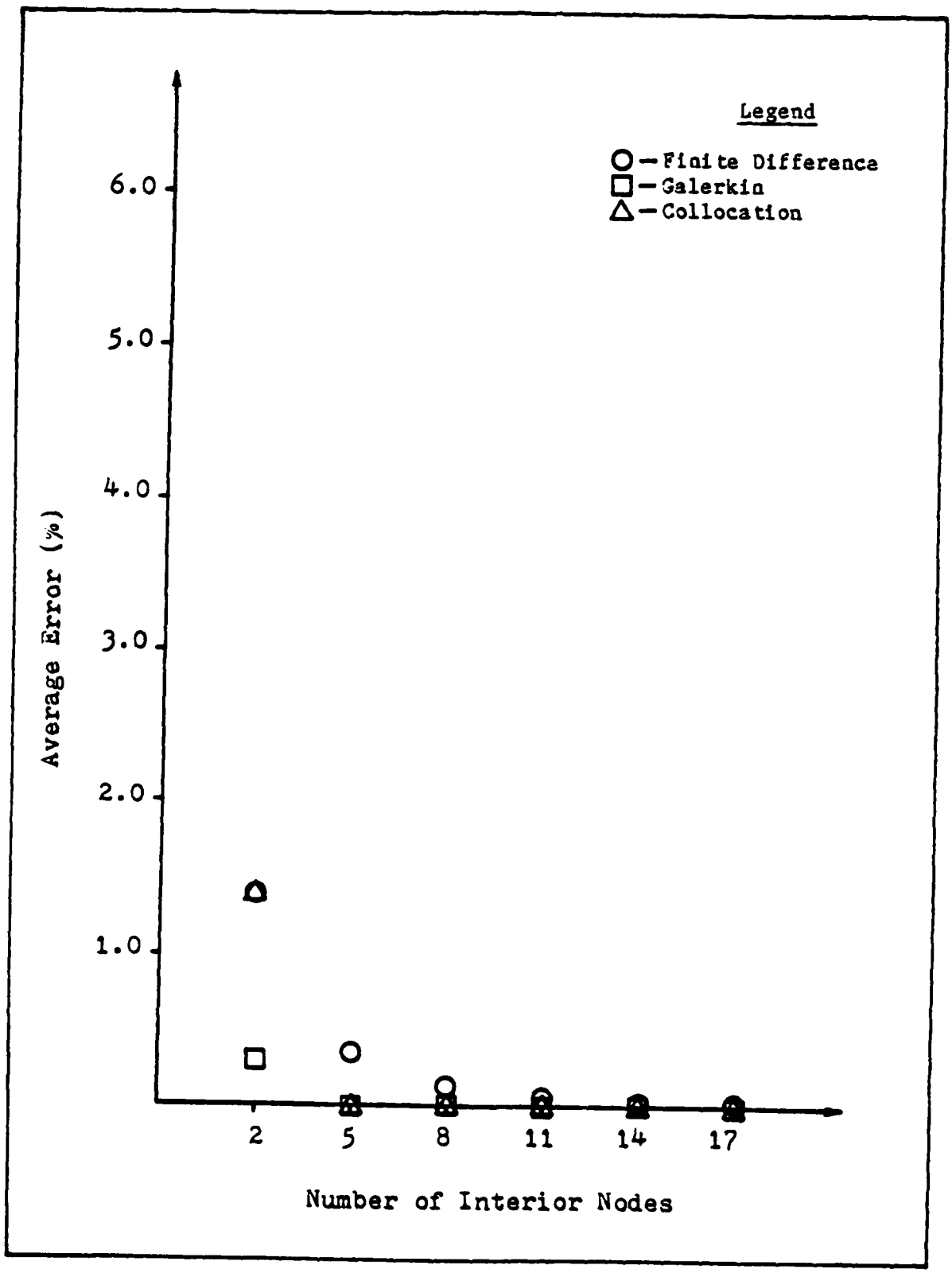


Figure 5. Average Error vs Number of Interior Nodes for $g(x) = x^2 + 1$

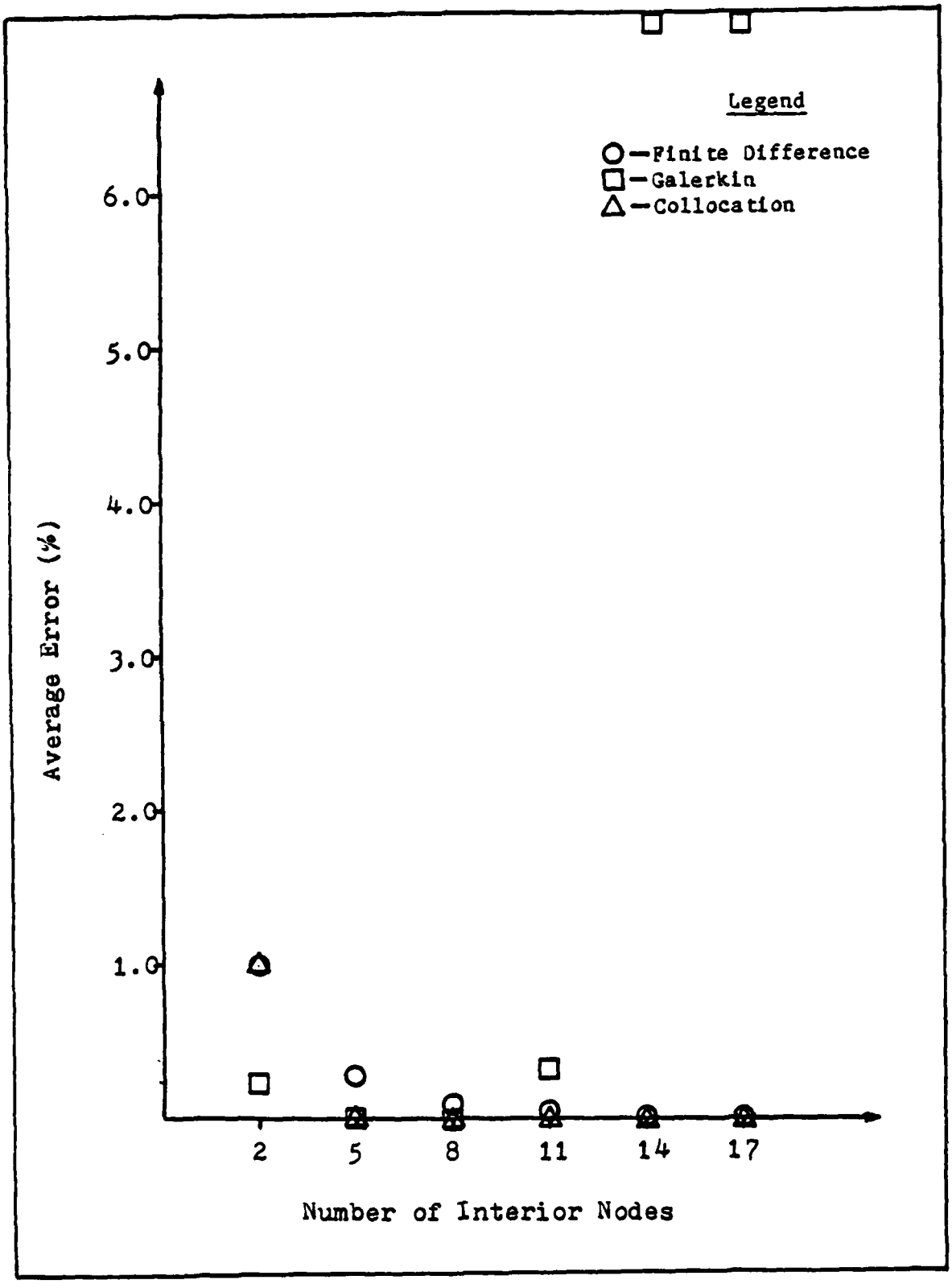


Figure 6. Average Error vs Number of Interior Nodes for $g(x) = x^2 + x + 1$

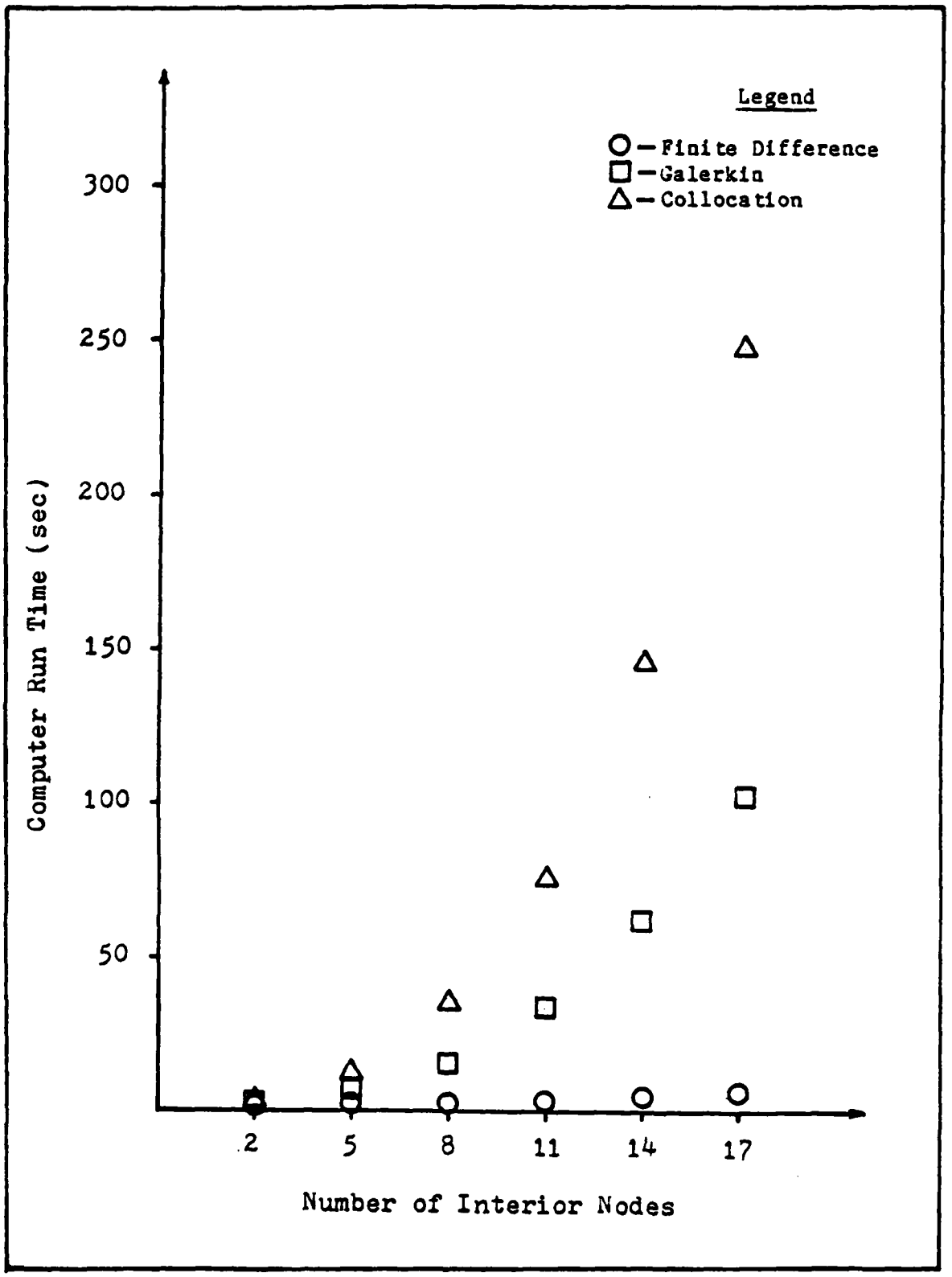


Figure 7. Computer Run Times vs Number of Interior Nodes

number of nodal points. The collocation program was the slowest of the three methods, largely due to the matrix solving technique used in this program. This program was based on an inversion routine using Gauss-Jordan elimination with column shifting (8:294-295), whereas the Galerkin program required the use (see Appendix A) of a direct Gaussian elimination routine with partial pivoting (9:192-193). Inversion methods require more computations to arrive at the solution than do direct methods, hence the large difference in the times between the two methods of weighted residuals.

Overall Solution Accuracy, and Comparison with Earlier Results. The results of this study do not support those reported in the earlier study (1). Clapp reported that in all cases, the finite difference method was superior to the method of weighted residuals for the one-dimensional case, yielding average errors of about one percent when 17 or more nodes were used. This study shows that both methods of weighted residuals achieved average errors on the order of 10^{-6} percent after only five interior nodes were used. This accuracy was matched by the finite difference method only for a constant inhomogeneity term.

None of the oscillations reported by Clapp for the collocation method were noted in this study.

Conclusions

From the analysis performed, it is clear that the method of weighted residuals is the best choice if few interior nodes are desired. While these methods are more difficult to program and take longer to run than the method of finite differences, the overall accuracy is much superior.

For the one-dimensional Poisson's equation, it appears that the collocation method is the more stable of the two methods of weighted residuals, and should therefore be the method of choice.

III. POISSON'S EQUATION IN TWO DIMENSIONS

The other problem examined in this study is the two-dimensional Poisson's equation. The general form of the problem can be expressed in the same manner as Eq(5)

$$L\, u(x,y) = g(x,y) \tag{50}$$

where L is now the two-dimensional linear differential operator, $d^2/dx^2 + d^2/dy^2$, $g(x,y)$ is the two-dimensional inhomogeneity term, and $u(x,y)$ is the unknown function to be determined. The Dirichlet boundary conditions associated with the two-dimensional problem are

$$u(0,y) = 0 \tag{51-a}$$

$$u(1,y) = 0 \tag{51-b}$$

$$u(x,0) = 0 \tag{51-c}$$

$$u(x,1) = 0 \tag{51-d}$$

Equations (50) and (51) define Poisson's equation for the region of a unit square.

Analytical Solution

The analytical solution to Eq (50), with its associated boundary conditions Eq (51), can be found by a Fourier series expansion (6:41-42) to be

$$u(x,y) = 4/\pi^2 \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \int_0^1 d\xi \int_0^1 d\eta \, \frac{\sin(m\pi x)\sin(n\pi y)\sin(m\pi\xi)\sin(n\pi\eta)g(\xi,\eta)}{(m^2 + n^2)} \tag{52}$$

For a given excitation, $g(x,y)$, the series solution to Eq (52) can be found by integration. This solution can then be programed on a

computer to provide a numerical solution, at a specific point, for the desired number of summation terms. It should be noted that because the solution involves a double summation process, obtaining the result may take a considerable length of time.

Numerical Approximations

For the two-dimensional case, the mesh is superimposed over a unit square, with s equally spaced nodes in the x direction, of step size h, and t equally spaced nodes in the y direction, of step size k. The total number of interior nodes in the mesh, N, is equal to s X t. A sample mesh for the two dimensional unit square, with the interior nodal points numbered, is shown in Figure 8.

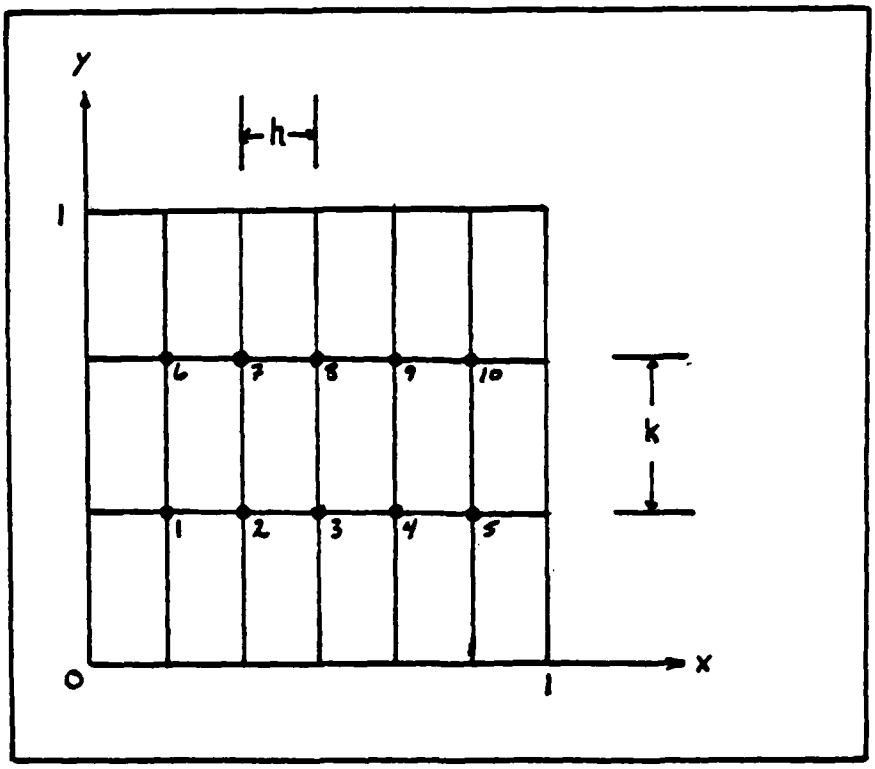


Figure 8. 2-D Mesh with 10 Interior Nodes

Since the number of interior nodal points used determines the number of simultaneous algebraic equations that must be solved, the x and y step sizes were chosen not to be equal. This was done in order to keep the matricies down to a manageable size.

Finite Difference Method. As was shown in section II (the one-dimensional Poisson's equation) the finite difference matricies, and the Green's function matricies were identical, therefore, the development of the finite difference method will only be done utilizing the Green's function approach.

The Green's function for the two dimensional linear operator, L, of Eq (50), is defined to be (15:1-18)

$$\partial^2 G(x|x_o;y|y_o)/\partial x^2 + \partial^2 G(x|x_o;y|y_o)/\partial dy^2 = \delta(x-x_o)\delta(y-y_o) \qquad (53)$$

with the associated boundary conditions (Eq(51)) being represented by

$$G(0|x_o;y|y_o) = 0 \qquad (54\text{-a})$$

$$G(1|x_o;y|y_o) = 0 \qquad (54\text{-b})$$

$$G(x|x_o;0|y_o) = 0 \qquad (54\text{-c})$$

$$G(x|x_o;1|y_o) = 0 \qquad (54\text{-d})$$

The discrete Green's function imposed on a mesh with step size h in the x direction, and k in the y direction, can be written as (4:315)

$$\partial^2 G_N(x|x_o;y|y_o)/\partial x^2 + \partial^2 G_N(x|x_o;y|y_o)/\partial y^2 = \delta(x-x_o)\delta(y-y_o)/hk \qquad (55)$$

The derivative terms can once again be replaced by central difference expressions

$$\partial^2 G_N(x|x_o;y|y_o)/\partial x^2$$
$$= (G_N(x+h|x_o;y|y_o) - 2G_N(x|x_o;y|y_o) + G_N(x-h|x_o;y|y_o))/h^2 \quad (56\text{-a})$$

and

$$\partial^2 G_N(x|x_o;y|y_o)/\partial y^2$$
$$= (G_N(x|x_o;y+k|y_o) - 2G_N(x|x_o;y|y_o) + G_N(x|x_o;y-k|y_o))/k^2 \quad (56\text{-b})$$

and then substituted back into Eq(55), so that the final form of the equation becomes

$$\begin{aligned} &k^2 G_N(x+h|x_o;y|y_o) - 2k^2 G_N(x|x_o;y|y_o) + k^2 G_N(x-h|x_o;y|y_o) \\ &+ h^2 G_N(x|x_o;y+k|y_o) - 2h^2 G_N(x|x_o;y|y_o) + h^2 G_N(x|x_o;y-k|y_o) \\ &= hk\,\delta(x-x_o)\delta(y-y_o) \end{aligned} \quad (57)$$

Applying this equation, and its associated boundary conditions (Eq(54)) at each of the N interior nodes of the mesh will result in a series of N^2 simultaneous algebraic equations. These equations can be represented in matrix notation as

$$\mathbf{C}\,\mathbf{G}_N = hk\mathbf{I}_N \quad (58)$$

where C is the coefficient matrix, $\mathbf{G}_N$ is the discrete Green's function matrix, and $\mathbf{I}_N$ is the identity matrix of order N. Eq(58) can be solved using matrix techniques to find $\mathbf{G}_N$, the approximation to the discrete Green's function matrix.

Method of Weighted Residuals. For the two-dimensional method of weighted residuals, the form of the solution is almost identical to the one dimensional case (Eq(16))

$$u(x,y) = \sum_n a_n b_n(x,y) \quad (59)$$

where the a_n are again constants, and the $b_n(x,y)$ are the basis functions, which were chosen to satisfy the boundary conditions (Eq(51)). Özisik (13:340-344) developed basis functions for a rectangular region as the product of a function f(x,y), and various powers of x and y. For the geometry of the problem in this study, the function f(x,y) was chosen to be

$$f(x,y) = xy(1-x)(1-y) \tag{60}$$

and the basis functions were chosen to be the same as those used by Clapp (1)

$$b_n = (xy)^{2(n-1)/3} f(x,y) \quad \text{(for } n=1,4,7,\ldots\text{)} \tag{61-a}$$

$$b_n = x^{2(n+1)/3} f(x,y) \quad \text{(for } n=2,5,8,\ldots\text{)} \tag{61-b}$$

$$b_n = y^{2n/3} f(x,y) \quad \text{(for } n=2,6,9,\ldots\text{)} \tag{61-c}$$

The basis functions were split into these three subgroups in order to facilitate the necessary integrations that follow in the sections on the Galerkin and collocation methods.

By substituting Eq(59) into Eq(50), the equation becomes

$$\sum_n a_n \, Lb_n(x,y) = g(x,y) \tag{62}$$

As in the one-dimensional case, a set of weighting functions, w_m, is then defined in the range of L, and the inner product of Eq(62) and these weighting functions is taken so that

$$\sum_n a_n \langle w_m, \, Lb_n(x,y)\rangle = \langle w_m, \, g(x,y)\rangle \qquad m=1,2,3,\ldots \tag{63}$$

which can again be represented in the more condensed matrix notation as

$$\mathbf{C}\,\mathbf{a} = \mathbf{g} \tag{64}$$

where C is the square coefficient matrix

$$\begin{bmatrix} \langle w_1,\ Lb_1(x,y)\rangle & \langle w_1,\ Lb_2(x,y)\rangle & \cdots \\ \langle w_2,\ Lb_1(x,y)\rangle & \langle w_2,\ Lb_2(x,y)\rangle & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{65}$$

a is the column vector

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix} \tag{66}$$

and g is the column vector

$$\begin{bmatrix} \langle w_1,\ g_1(x,y)\rangle \\ \langle w_2,\ g_2(x,y)\rangle \\ \vdots \end{bmatrix} \tag{67}$$

The values of a can then be computed from

$$\mathbf{a} = \mathbf{C}^{-1}\,\mathbf{g} \tag{68}$$

and the approximate solutions can then be written as

$$u(x,y) = \mathbf{b}^{\dagger}\,\mathbf{a} \tag{69}$$

for a method valid over the entire region, or as

$$\mathbf{u} = \mathbf{B}\,\mathbf{a} \tag{70}$$

for a method valid only at discrete points, where

$$\mathbf{b}^t = [b_1,\ b_2,\ b_3,\ \ldots,\ b_N] \tag{71}$$

$$\mathbf{u} = \begin{bmatrix} u(x_1,y_1) \\ u(x_2,y_2) \\ \vdots \\ u(x_N,y_N) \end{bmatrix} \tag{72}$$

and

$$\mathbf{B} = \begin{bmatrix} b_1(x_1,y_1),\ b_2(x_1,y_1),\ \ldots,\ b_N(x_1,y_1) \\ b_1(x_2,y_2),\ b_2(x_2,y_2),\ \ldots,\ b_N(x_2,y_2) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ b_1(x_N,y_N),\ b_2(x_N,y_N),\ \ldots,\ b_N(x_N,y_N) \end{bmatrix} \tag{73}$$

Galerkin's Method. In the two-dimensional method of Galerkin, the weighting functions are again chosen as being equal to the basis functions Eq(61). The values for the coefficient matrix, C can be found by taking the inner product of w_m and $L\,b_n(x,y)$, resulting in

$$C_{mn} = \langle w_m,\ Lb_n(x,y)\rangle = \int_0^1\int_0^1 w_m\ (d^2/dx^2 + d^2/dy^2)\ b_n(x,y)\ dx\ dy \tag{74}$$

and g_m can be found in a similar manner to be

$$g_m = \langle w_m,\ g(x,y)\rangle$$

$$= \int_0^1\int_0^1 w_m\ (d^2/dx^2 + d^2/dy^2)\ (Ax^2 + By^2 + Cx + Dy + E)\ dx\ dy \tag{75}$$

where in each case, w_m is defined to be equal to $b_n(x,y)$ (Eq(61)) for the various values of n.

There are no restrictions on the value of x or y in Eqs(74) and (75), so the Galerkin approximations are valid over the entire region.

Collocation Method. In the two-dimensional method of collocation, the weighting functions are chosen to be equal to the two-dimensional Dirac delta function

$$w_m = \delta(x-x_m)\delta(y-y_m) \tag{76}$$

where the coordinate x_m is defined to be the x coordinate of the mth interior node and y_m is defined to be the y coordinate of the mth interior node. The values for the coefficient matrix, **C** can be found by taking the inner product of w_m and L $b_n(x,y)$

$$\begin{aligned} C_{mn} &= \langle w_m,\ Lb_n(x,y)\rangle \\ &= \int_0^{l'}\int_0^{l'} \delta(x-x_m)\delta(y-y_m)\ (d^2/dx^2 + d^2/dy^2)\ b_n(x,y)\ dx\ dy \end{aligned} \tag{77}$$

and g_m can be found to be

$$\begin{aligned} g_m &= \langle w_m,\ g(x,y)\rangle \\ &= \int_0^{l'}\int_0^{l'} \delta(x-x_m)\delta(y-y_m)\ (d^2/dx^2 + d^2/dy^2)\ (Ax^2 + By^2 + Cx \\ &\qquad + Dy + E)\ dx\ dy \end{aligned} \tag{78}$$

The values of x and y are restricted to the coordinates of the interior nodal points for the collocation method, therefore, the approximations are only valid at these points, and not over the entire region.

Green's Functions and Analogs

Since the finite difference method was developed in terms of the Green's function, only the method of weighted residuals will be addressed in this section.

For the two-dimensional Poisson's equation (Eq(50)), and its associated boundary conditions (Eq(51)), the Green's function for the problem can be determined analytically (6:42-43). The solution to Eq(50) with its various inhomogeneity terms can be found by calculating the integral

$$u(x,y) = \int_0^1 \int_0^1 G(x|x_o;y|y_o)\ g(x,y)\ dx_o\ dy_o \tag{79}$$

where $G(x|x_o;y|y_o)$ is the Green's function for Eq(50) and its associated boundary conditions, x and y are the field point coordinates, and x_o and y_o are the source point coordinates.

Eq(79) may be written in matrix notation, for the discrete Green's function on a mesh of step sizes h_s in the x direction, and k_t in the y direction, as

$$\tilde{\mathbf{u}}_N = hk\ \mathbf{G}_N\ \mathbf{g}_N \tag{80}$$

where $\tilde{\mathbf{u}}$ is the column vector of solutions at discrete points on the mesh for the given inhomogeneity, and the tilde again signifies that the discrete Green's function solution may not necessarily be equal to the solution obtained by the method of weighted residuals, $\mathbf{u}$.

The two-dimensional forms of Eqs(42) and (43) are the approximate solutions

$$u_N = \mathbf{b}^t\ \mathbf{C}^{-1}\ \mathbf{g} \tag{81}$$

and

$$u_N = B\, C^{-1}\, g \tag{82}$$

Again, the weighted residual approximations do not contain the factors for the step sizes h and k. This is due to the fact that the weighted residual approximations involve a summation over N terms, while the discrete Green's function solution was developed by approximating an integral (Eq(79)), where the step sizes h and k correspond to the dx and dy terms. The analog to the discrete Green's function in Eqs(81) and (82) can be defined to be

$$\bar{G}_N^* = \bar{b}^t\, C^{-1} \tag{83}$$

for a method valid over the entire region, such as Galerkin's method (where the bar notation again indicates a column vector, and was used to prevent confusion with the inhomogeneity term), and

$$G_N^* = B\, C^{-1} \tag{84}$$

for a method valid only at discrete points, such as the collocation method. The asterisk indicates that they are analogs to the discrete Green's function, and that the elements that make up the inhomogeneity vector, g, may not necessarily be equal to the inhomogeneity term g(x,y) because of the weighting factor.

Computer Analysis

For the two-dimensional case, new programs were developed from the one-dimensional programs to handle the approximations for the three techniques. The program listings are included in Appendix C.

Inhomogeneity Terms and Exact Solutions. The same four excitation terms chosen by Clapp (1) were used in the two-dimensional case so that comparisons could be made with the results obtained in his study. The four problems were

$$1) \quad L\,u(x) = 10 \tag{85-a}$$

$$2) \quad L\,u(x) = x^2 \tag{85-b}$$

$$3) \quad L\,u(x) = x^2 + y^2 \tag{85-c}$$

$$4) \quad L\,u(x) = x^2 + y^2 + x \tag{85-d}$$

with the associated Dirichlet boundary conditions

$$u(0,y) = 0 \tag{86-a}$$

$$u(1,y) = 0 \tag{86-b}$$

$$u(x,0) = 0 \tag{86-c}$$

$$u(x,1) = 0 \tag{86-d}$$

The number of interior nodes were carefully chosen so that the solution comparisons could all be made at the same four x,y nodal points; (1/3,1/3), (2/3,1/3), (1/3,2/3), and (2/3,2/3).

The analytical solutions to the problem set were found by integrating Eq(52) with a general form of the equations used for the inhomogeneity term,

$$g(x,y) = Ax^2 + By^2 + Cx + Dy + E \tag{87}$$

where A, B, C, D and E are all constants.

The solution for the general inhomogeneity term with the boundary conditions of Eq(86) is

$$u(x,y) = -4/\pi^2 \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} [\sin(m\pi x)\sin(n\pi y)/(m^2 + n^2)]$$
$$\{A[(2/m^3\pi^3 - 1/m\pi)(-1)^m - 2/m^3\pi^3]\ (1/n\pi)(1 - (-1)^n)$$
$$+ B[(2/n^3\pi^3 - 1/n\pi)(-1)^n - 2/n^3\pi^3]\ (1/m\pi)(1 - (-1)^m) \quad (38)$$
$$- C[(-1)^m/mn\pi^2](1 - (-1)^n) - D[(-1)^n/mn\pi^2](1 - (-1)^m)$$
$$+ E[1/mn\pi^2]((-1)^m - 1)((-1)^n - 1)\}$$

This solution was then programed on the computer, and the numerical values for the inhomogeneities at each point were generated by summing over both m and n from one to seventy. Each numerical value then was obtained using 4900 summation terms. The numerical values for the four problems are listed in Table 2.

TABLE 2

Exact Solutions to Eq(38)

Problem #	(1/3,1/3)	(2/3,1/3)	(1/3,2/3)	(2/3,2/3)
1	-.6034615	-.6034615	-.6034615	-.6034615
2	-.0126051	-.0230496	-.0126051	-.0230496
3	-.0252102	-.0356547	-.0356547	-.0460993
4	-.0501611	-.0710501	-.0606056	-.0814946

Average Error. For the two-dimensional case, the average percent error was defined to be

$$\langle E_N \rangle = \left\{ \frac{|u_N(\text{point i}) - u(\text{point i})|}{u(\text{point i})} \right\} \cdot \frac{100}{4} \quad (39)$$

where u_N is the approximation at a specific point, and u is the exact solution.

Comparison of Approximations to the Exact Solutions. The plots of the average percent error vs number of interior nodal points for each of the four equations in the problem set are shown in Figures 9-12.

As in the earlier study (1), the collocation method failed to converge to the correct solution despite using a direct Gaussian elimination routine instead of performing a matrix inversion. A more detailed discussion of the problem with the collocation method can be found in Appendix A.

In each of the four cases for the other two techniques, the Galerkin method yielded better results than the finite difference method, which is not only consistent with the results reported for the one-dimensional case, but also with the results reported by Clapp.

Computer Run Times. Each program was timed by hand using an electronic stop-watch to obtain values for the computer run time. The times were again averaged, since they differed by only one or two seconds. The plot of program run time vs number of interior nodal points is shown in Figure 13.

In all cases, the method of finite differences was again the quickest of the three programs to run. While the two-dimensional finite difference coefficient matrix was not a tri-diagonal matrix (as in the one-dimensional case), it was still a relatively sparse one (ie. few non-zero terms). Because there were fewer computations required to create the finite difference coefficient matrix than for the other two methods, it ran faster (despite being based on a matrix inversion routine). Both methods of weighted residuals were developed using the direct Gaussian elimination routine.

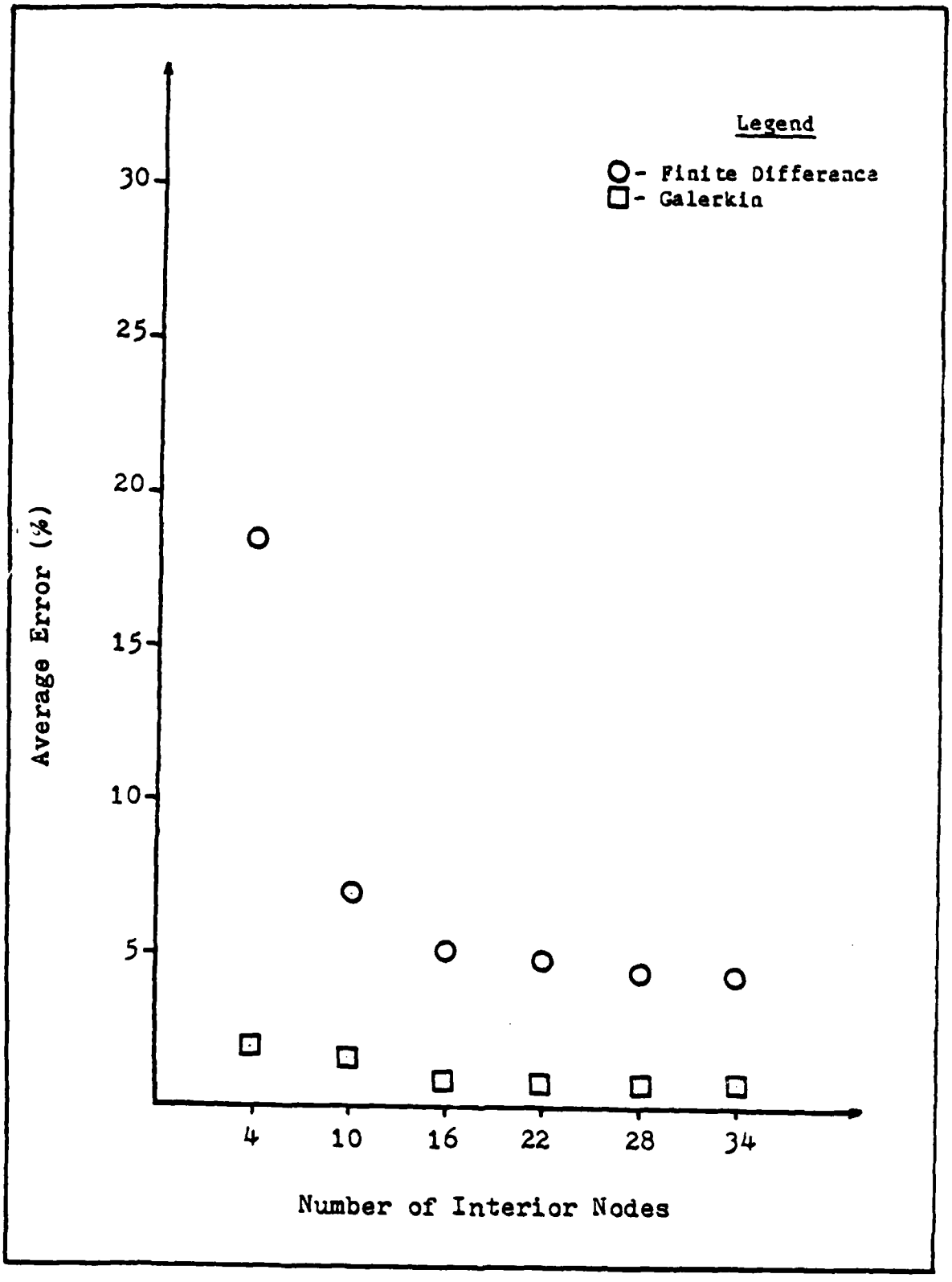


Figure 9. Average Error vs Number of Interior Nodes for g(x,y) = 10

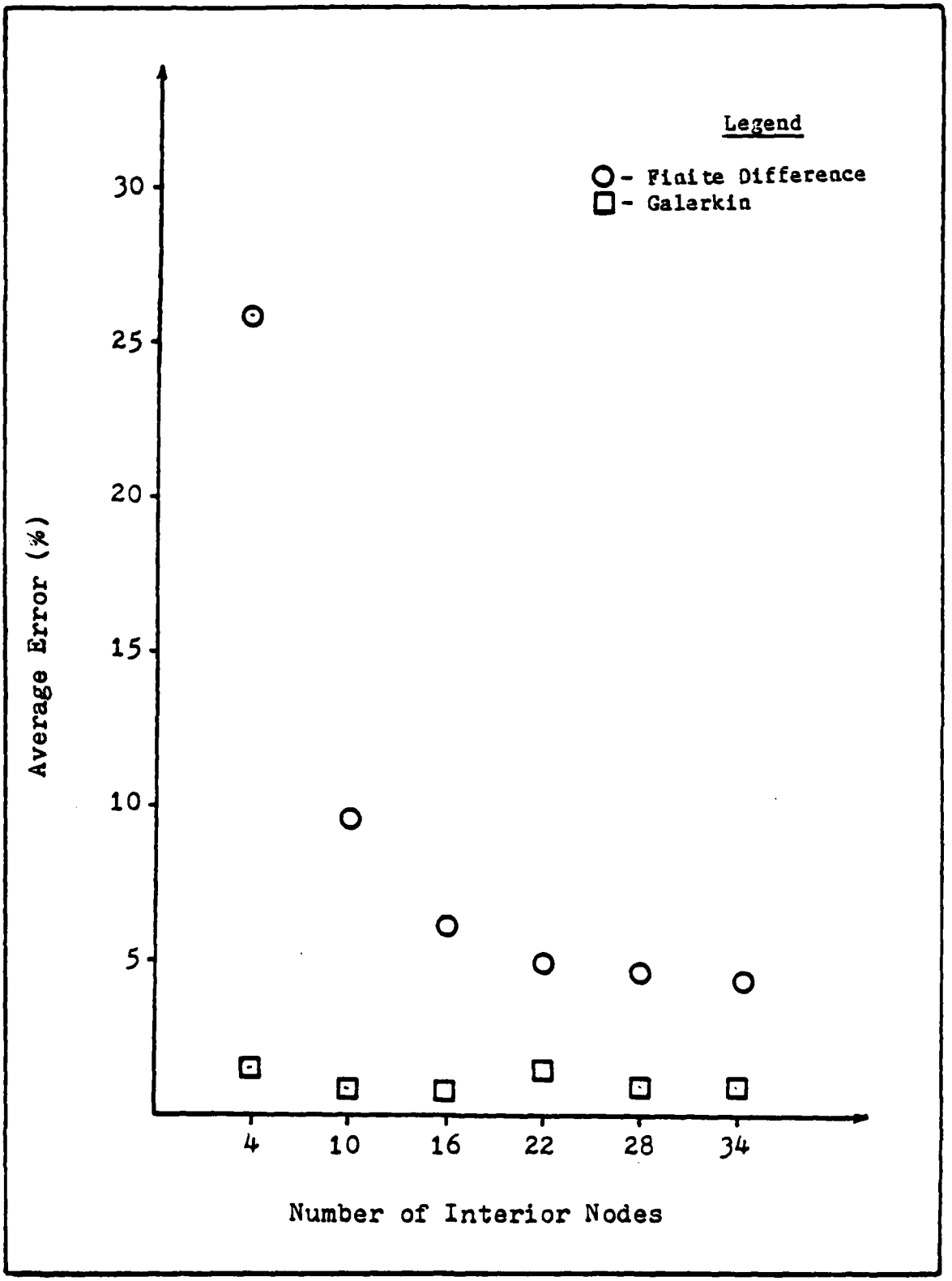


Figure 10. Average Error vs Number of Interior Nodes for $g(x,y) = x^2$

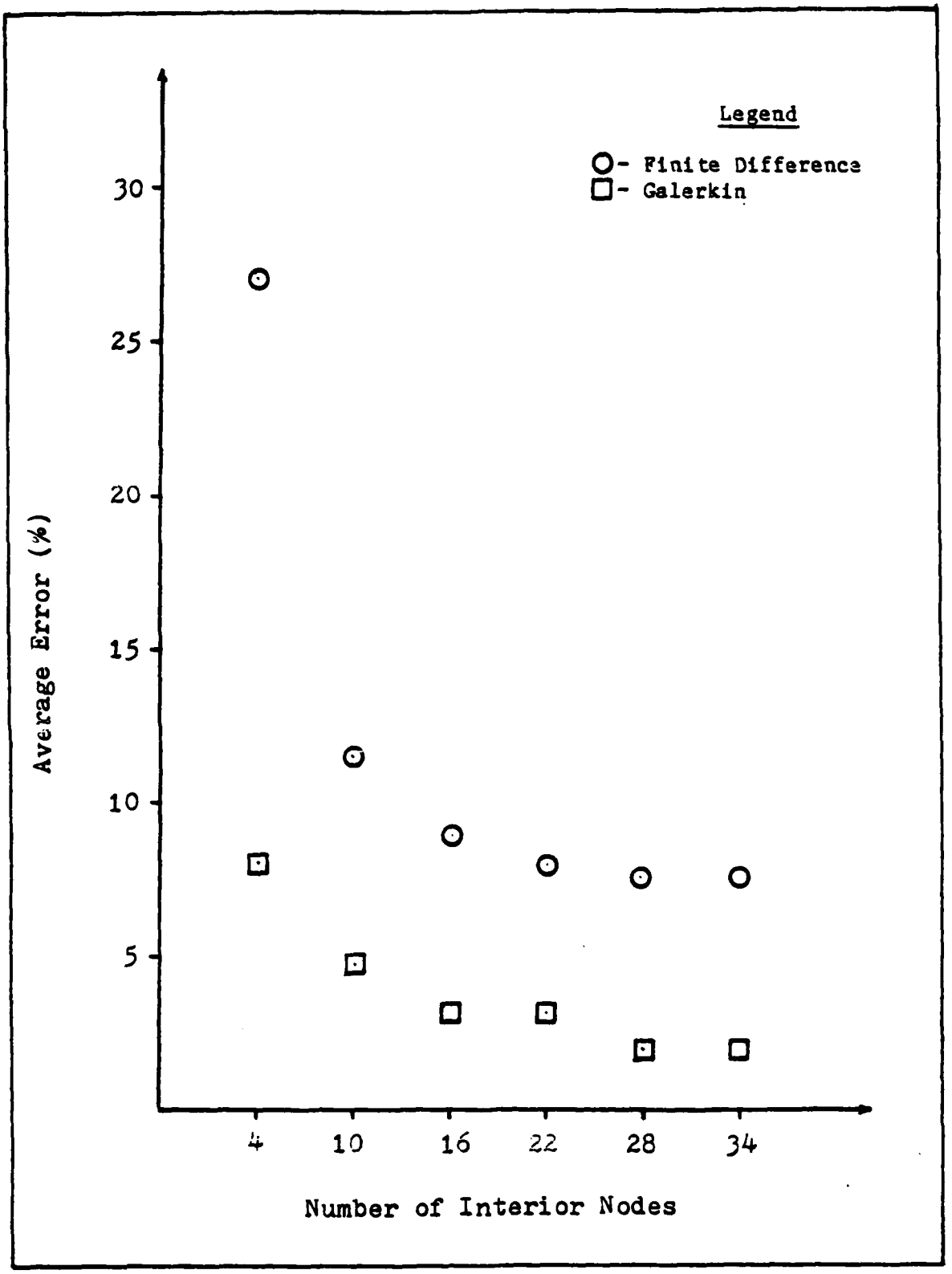


Figure 11. Average Error vs Number of Interior Nodes for $g(x,y) = x^2 + y^2$

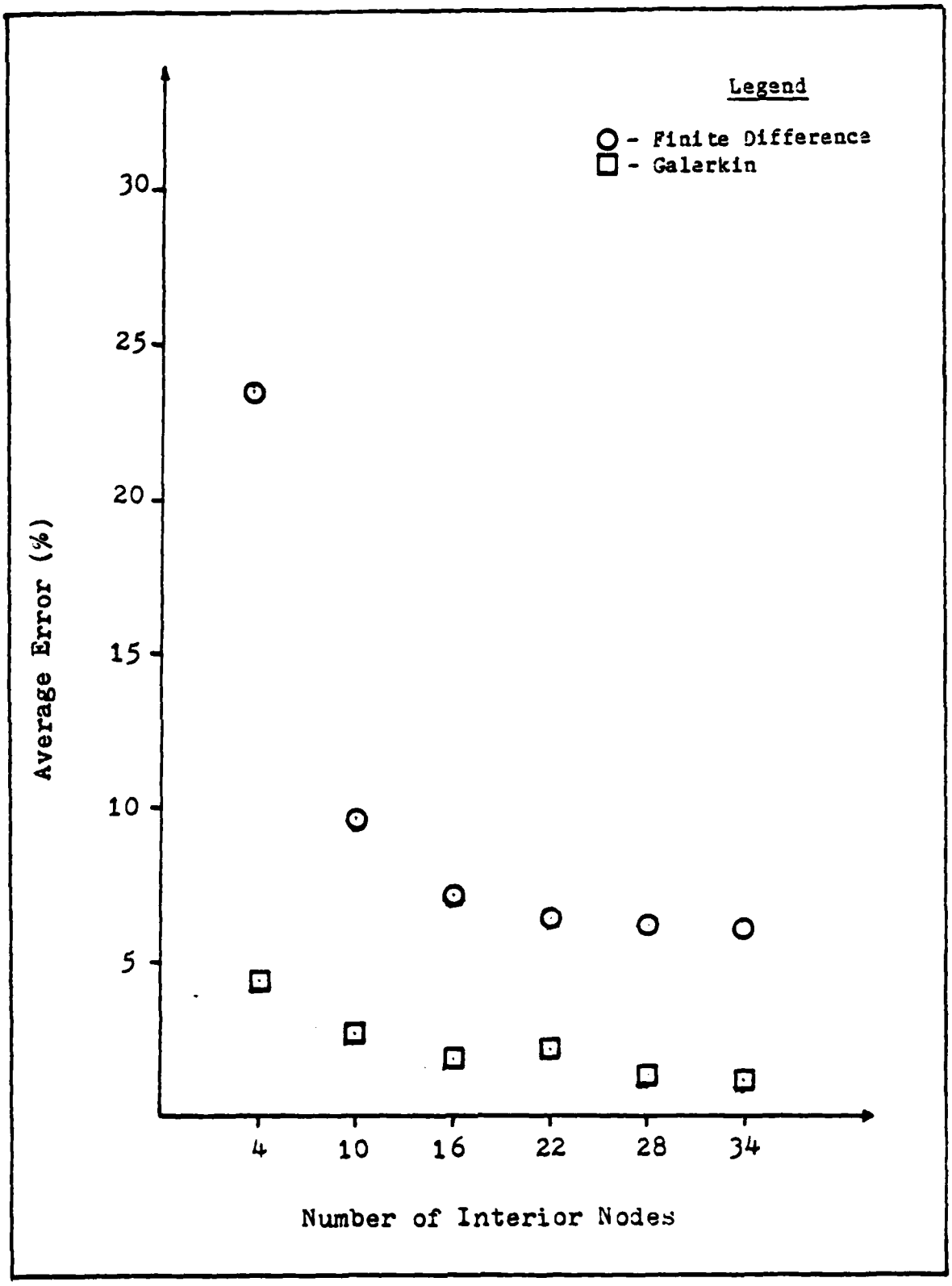


Figure 12. Average Error vs Number of Interior Nodes for $g(x,y) = x^2 + y^2 + x$

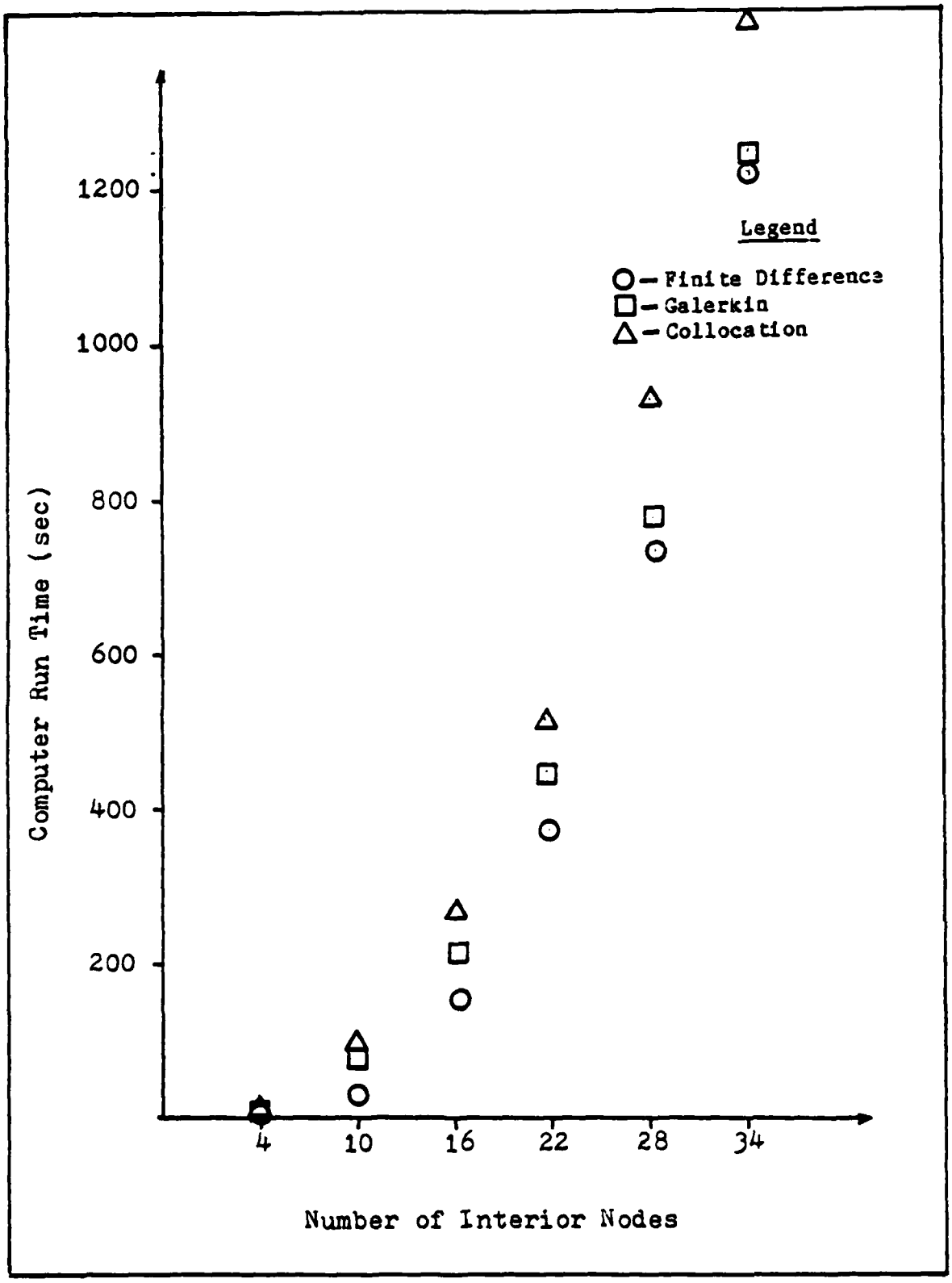


Figure 13. Computer Run Times vs Number of Interior Nodes

Overall Solution Accuracy, and Comparison with Earlier Results. The results of this section of the study support those reported by Clapp (1), the Galerkin method yielded better results than did the finite difference method, with initial average errors from 1.5 to 3.0 percent, compared with 18.5 to 27.0 percent. The Galerkin method converged to within one percent after 34 interior nodes were used, while the finite difference method remained above four percent.

A slight oscillation in the Galerkin results was noticed for 22 interior nodal points.

Conclusions

From the analysis performed, the method of weighted residuals was once again superior to the method of finite differences, for fewer interior nodal points.

Neither method was as accurate as its one-dimensional counterpart. This is most likely due to round off error caused by the larger matrix sizes, and to the rather lengthy recursion relations used to create the coefficient matricies.

While the Galerkin method was clearly better than the finite difference method, it is also more difficult and tedious to change the inhomogeneity term, or the form of the differential equation, due to the integrals that must be solved. Despite this difficulty, the Galerkin approach is clearly the method of choice for solving the two-dimensional Poisson's equation.

IV. Conclusions and Recommendations

Conclusions

Several points should be made concerning the use of the method of weighted residuals for determining approximations to the discrete Green's function.

First, both the Galerkin and collocation methods yield analogs of the Green's function, which are as useful as the discrete Green's function itself, and they can be used (at least theoretically) to find the solution to the one- and two-dimensional Poisson's equation with various inhomogeneity terms. A major drawback to the use of the method of weighted residuals is that the inhomogeneity matrix must be recalculated for each different inhomogeneity term. This involves the calculation of a lengthy double integration in the two-dimensional problem, especially in the case of Galerkin's method.

The next point is the criticality of the choice of basis functions. Both the one-dimensional Galerkin routine, and the two-dimensional collocation routine seemed to show the effects of the choice of basis functions resulting in ill-conditioned matricies.

Finally, the method of weighted residuals takes more time to run on the computer than does the finite difference method. For low numbers of interior nodes, this may not be much of a problem, but in the two-dimensional programs, calculations involving 34 nodal points took 25 minutes to arrive at a solution.

Recommendations

The first step in any follow-on study should be the actual calculation of the Green's function for the various methods, so that

they can be compared for accuracy.

In addition, one other major area requires further study in utilizing the method of weighted residuals - the choice of basis functions. An in-depth study of the orthogonal collocation method, and the choice of basis functions in general would be most beneficial.

The choice of using a micro-computer for developing the programs was probably not very wise. Although the results were as accurate as those done on mainframe computers, the programs took too long to run. Future work should be done on a larger, faster machine.

Bibliography

1. Clapp, 1/Lt Randolph E. Investigation of the Numerical Method of Moments for Digital Computer Determination of Green's Functions. MS Thesis AFIT/GEP/PH/83D-3. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, October 1983.

2. Conte, Samuel D. and C. de Boor. Elementary Numerical Analysis. New York: McGraw-Hill Book Company, 1972.

3. Edwards, Capt Thomas Y. The Solution of Mixed Boundary Value Problems Using Numerical Green's Functions. MS Thesis GEP/PH/71-5. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, June 1971.

4. Finlayson, Bruce A. The Method of Weighted Residuals and Variational Principles. New York: Academic Press, 1972.

5. Forsythe, George E. and W. R. Wasow. Finite-Difference Methods for Partial Differential Equations. New York: John Wiley and Sons, Inc., 1960.

6. Gallof, 1/Lt Sanford. A Numerical Investigation and Utilization of Green's Functions. MS Thesis GA/PH/68-2. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, June 1968.

7. Harrington, R. F. Field Computation by Moment Methods. New York: The Macmillan Company, 1968.

8. Hornbeck, Robert W. Numerical Methods. New York: Quantum Publishers, Inc., 1975.

9. James, Merlin L., G. M. Smith and J. C. Wolford. Applied Numerical Methods for Digital Computation with FORTRAN. Scranton, PA: International Textbook Company, 1967.

10. Knecht, Ken. Microsoft Basic. Beaverton, OR: dilithium Press, 1979.

11. Martin, Harold C. and G. F. Carey. Introduction to Finite Element Analysis. New York: McGraw-Hill Book Company, 1973.

12. Naegeli, Capt Robert E. Investigation of the Numerical Methods of Finite Differences and Weighted Residuals for the Solution of the Heat Equation. MS Thesis AFIT/GNE/PH/81-7. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, March 1981.

13. Ozisik, M. Necati. Boundary Value Problems of Heat Conduction. Scranton, PA: International Textbook Company, 1968.

14. Smith, Gordon D. Numerical Solution of Partial Differential Equations. New York: Oxford University Press, 1965.

15. Stakgold, Ivar. Boundary Value Problems of Mathematical Physics. Vol. 1. New York: The Macmillan Company, 1967.

Appendix A

Ill-conditioned Matricies

The divergence from the correct solution reported for the one-dimensional Galerkin routine with $g(x) = x^2 + x + 1$, and for the two-dimensional collocation routine, were both the results of ill-conditioned matricies.

Ill-conditioned matricies are nearly singular systems (ie. those that have no unique solution) which are extremely sensitive to small changes in the coefficient matrix, C, and the right hand side, g. For example, the solutions to the following two similar matrix equations differ greatly:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1.00001 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{90}$$

and

$$\begin{bmatrix} 1 & -1 \\ -1 & 0.99999 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{91}$$

The solution to Eq(90) is [10001 100000], and the solution to Eq(91) is [-99999 -100000] (11:343). This shows that a relatively small change in the value of C_{22} by .00002 can cause a large change in the solution.

It is easiest to interpret what is happening geometrically. Each solution may be thought of as representing the point of intersection of two nearly parallel lines. Any slight shift in either of the two lines will greatly change the point of intersection.

Ill-conditioning of a system may be attributed to any of the following sources (11:345-347):

1. For well-conditioned physical problems, ill-conditioned equations may be caused by a correct, but very fine mesh idealization. No numerical problems are encountered when the problem is solved with coarse idealizations. As the mesh is repeatedly subdivided, the condition number increases. Eventually the buildup of error due to round off in the calculations swamps any accuracy improvement due to the finer discretization.

2. The form of the right-hand-side vector can have a significant effect in many applications. For example, in the bending and stretching of a flat plate, the stiffness matrix may uncouple into an ill-conditioned submatrix and a well-conditioned submatrix. In complex structural systems, weak coupling can occur so that a force vector acting on the ill-conditioned part will excite the ill-conditioning; whereas if it acts on the well-conditioned part, highly accurate solutions are produced.

3. The condition of the system is influenced by the choice of basis functions.

The two-dimensional collocation routine seemed to exhibit the traits listed in source 1. The program was run for different numbers of interior nodal points than the ones used for the other programs. Figure 14 shows a plot of the average percent error vs number of interior nodal points for the collocation method when fewer nodal points were used. The plot shows that the collocation method was in fact converging to the correct solution until the mesh size passed a critical value.

The one-dimensional Galerkin routine seemed to exhibit the traits of source 2. It worked perfectly well for all other inhomogeneity terms, but failed for $g(x) = x^2 + x + 1$ beyond 11 interior nodal points.

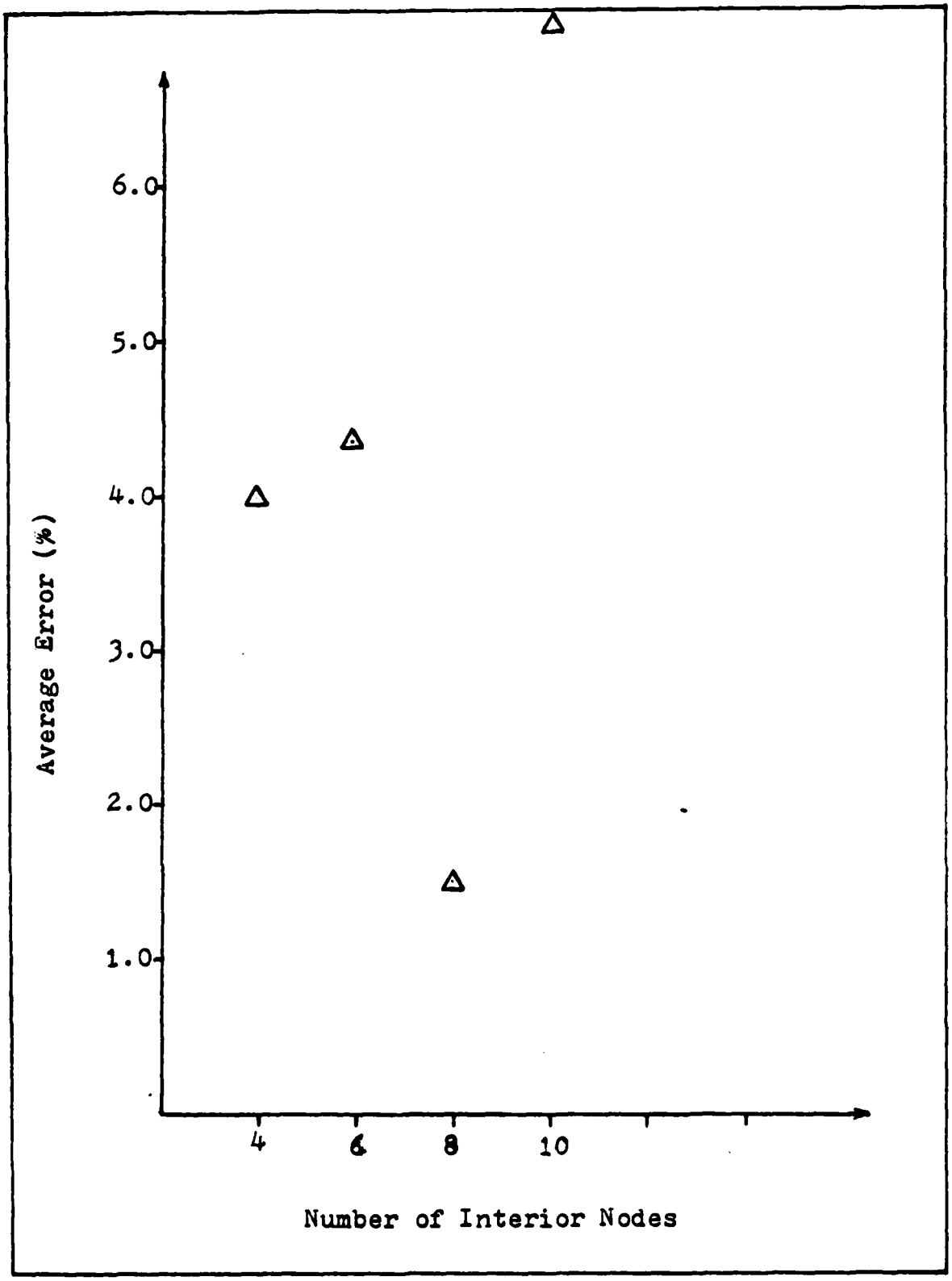


Figure 14. Average Error vs Number of Interior Nodes
2-D Collocation Routine for $g(x,y) = x^2$

One way of determining if a matrix is ill-conditioned is to observe whether the condition number increases significantly as the size of the matrix is increased. The condition number for a matrix can be calculated by multiplying the value of $C_{i,max}$ by $C^{-1}_{i,max}$, where $C_{i,max}$ is defined to be the summation of all of the terms within the column of the matrix leading to the greatest value, and $C^{-1}_{i,max}$ is the same calculation performed on the inverse of the matrix.

Table 3 shows how the condition number for the two-dimensional collocation coefficient matrix increased as the size of the matrix was increased.

Table 3

Condition Numbers for the 2-d Collocation Matrix

Size of Matrix	Condition Number
4 X 4	2.7×10^{1}
6 X 6	5.9×10^{2}
8 X 8	1.6×10^{3}
10 X 10	2.6×10^{5}
20 X 20	3.9×10^{12}

Little can be done to improve the solutions obtained from an ill-conditioned matrix. One improvement though is to use a direct method of solving the matrix, rather than an inversion routine. This was done for the one-dimensional Galerkin program when it was discovered that the approximations were all diverging from the correct solution after only eight interior nodes. Once the matrix inversion routine was replaced with a direct Gaussian elimination routine, the results improved significantly, except in the aforementioned case.

Changing the matrix solving routine was not enough however to make a difference in the two-dimensional collocation program. Others (11:345-347, 12:60-65) have suggested that the ill-conditioning may be the result of the improper choice of basis functions; that orthogonal polynomials would be better suited for use in this situation. A detailed discription of the orthogonal collocation method can be found in Finlayson (4:97-107).

Appendix B

Numerical Approximations for the Problem Sets

This Appendix contains the values for the finite difference, Galerkin, and collocation approximations to Eq(46) and the finite difference and Galerkin approximations to Eq(85) (the results of the two-dimensional collocation routine are discussed in Appendix A).

The values listed are the outputs from the computer routines listed in Appendix C, rounded to seven decimal places. These values are listed by inhomogeneity term within each method, and are tabulated according to the number of interior nodes used. The average percent error is also listed for each set of values.

Table 4

Finite Difference Approximations for $g(x) = 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-1.1111111	-1.1111111	0.0
5	-1.1111109	-1.1111109	1.1 E-5
8	-1.1111111	-1.1111111	0.0
11	-1.1111113	-1.1111113	2.1 E-5
14	-1.1111110	-1.1111110	6.8 E-6
17	-1.1111108	-1.1111108	2.8 E-5

Table 5

Finite Difference Approximations for $g(x) = x^2$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.0246913	-.0370370	6.5
5	-.0262346	-.0385802	1.6
8	-.0265203	-.3886602	0.7
11	-.0266204	-.0389661	0.4
14	-.0266666	-.0390123	0.3
17	-.0266918	-.0390375	0.2

Table 6

Finite Difference Approximations for $g(x) = x^2 + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1358025	-.1481481	1.4
5	-.1373457	-.1496913	0.4
8	-.1376315	-.1499771	0.2
11	-.1377315	-.1500772	0.1
14	-.1377777	-.1501234	6.0 E-2
17	-.1378029	-.1501486	4.0 E-2

Table 7

Finite Difference Approximations for $g(x) = x^2 + x + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1851852	-.2098765	1.0
5	-.1867284	-.2114197	0.3
8	-.1870142	-.2117055	0.1
11	-.1871142	-.2118056	6.0 E-2
14	-.1871605	-.2118518	4.0 E-2
17	-.1871856	-.2118769	3.0 E-2

Table 8

Galerkin Approximations for $g(x) = 10$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-1.1111111	-1.1111112	6.1 E-6
5	-1.1111111	-1.1111112	6.1 E-6
8	-1.1111111	-1.1111112	6.1 E-6
11	-1.1111111	-1.1111112	6.1 E-6
14	-1.1111111	-1.1111112	6.0 E-6
17	-1.1111111	-1.1111112	6.1 E-6

Table 9

Galerkin Approximations for $g(x) = x^2$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.0271605	-.0395062	1.3
5	-.0267490	-.0390947	3.5 E-6
8	-.0267490	-.0390947	3.5 E-6
11	-.0267490	-.0390947	3.5 E-6
14	-.0267490	-.0390947	3.5 E-6
17	-.0267490	-.0390947	3.5 E-6

Table 10

Galerkin Approximations for $g(x) = x^2 + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1382716	-.1506173	0.3
5	-.1378601	-.1502058	5.4 E-6
8	-.1378601	-.1502058	5.4 E-6
11	-.1378601	-.1502058	5.4 E-6
14	-.1378601	-.1502058	5.4 E-6
17	-.1378601	-.1502058	5.4 E-6

Table 11

Galerkin Approximations for $g(x) = x^2 + x + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1876543	-.2123457	0.2
5	-.1872426	-.2119341	1.7 E-5
8	-.1872351	-.2119680	1.0 E-2
11	-.1884510	-.2118339	0.3
14	-.2158807	.1950268	103.7
17	-.2414421	.1324444	95.7

Table 12

Collocation Approximations for $g(x) = 10$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-1.1111111	-1.1111112	6.1 E-6
5	-1.1111111	-1.1111112	6.1 E-6
8	-1.1111111	-1.1111112	6.1 E-6
11	-1.1111111	-1.1111112	6.1 E-6
14	-1.1111111	-1.1111112	6.2 E-6
17	-1.1111111	-1.1111113	3.8 E-6

Table 13

Collocation Approximations for $g(x) = x^2$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.0246914	-.3703704	6.5
5	-.0267490	-.0390947	3.5 E-6
8	-.0267490	-.0390947	3.5 E-6
11	-.0267490	-.0390947	3.5 E-6
14	-.0267490	-.0390947	3.5 E-6
17	-.0267490	-.0390947	6.1 E-6

Table 14

Collocation Approximations for $g(x) = x^2 + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1358025	-.1481482	1.4
5	-.1378601	-.1502058	5.4 E-6
8	-.1378601	-.1502058	5.4 E-6
11	-.1378601	-.1502058	5.4 E-6
14	-.1378601	-.1502058	5.4 E-6
17	-.1378601	-.1502058	7.6 E-6

Table 15

Collocation Approximations for $g(x) = x^2 + x + 1$

Number of Interior Nodes	x = 1/3	x = 2/3	Average Error (%)
2	-.1851852	-.2098766	1.0
5	-.1872428	-.2119342	5.4 E-6
8	-.1872428	-.2119342	5.4 E-6
11	-.1872428	-.2119342	5.9 E-6
14	-.1872428	-.2119342	5.1 E-6
17	-.1872428	-.2119341	8.9 E-6

Table 16

Finite Difference Approximations for $g(x,y) = 10$

Number of Interior Nodes	$x = 1/3$ $y = 1/3$	$x = 2/3$ $y = 1/3$	$x = 1/3$ $y = 2/3$	$x = 2/3$ $y = 2/3$	Average Error (%)
4	-.5128205	-.4059829	-.5341880	-.5128205	13.5
10	-.5653312	-.5469123	-.5684183	-.5640845	7.0
16	-.5734571	-.5674212	-.5744274	-.5729557	5.2
22	-.5759343	-.5732613	-.5763574	-.5756971	4.7
28	-.5769715	-.5755534	-.5771928	-.5768423	4.4
34	-.5774941	-.5766637	-.5776240	-.5774169	4.3

Table 17

Finite Difference Approximations for $g(x,y) = x^2$

Number of Interior Nodes	$x = 1/3$ $y = 1/3$	$x = 2/3$ $y = 1/3$	$x = 1/3$ $y = 2/3$	$x = 2/3$ $y = 2/3$	Average Error (%)
4	-.0092593	-.0146605	-.0100309	-.0185135	25.8
10	-.0114660	-.0202691	-.0116141	-.0210928	9.4
16	-.0118848	-.0213449	-.0119352	-.0216323	6.1
22	-.0120202	-.0216807	-.0120430	-.0218120	5.1
28	-.0120784	-.0218200	-.0120906	-.0218904	4.6
34	-.0121082	-.0218888	-.0121155	-.0219309	4.4

Table 18

Finite Difference Approximations for $g(x,y) = x^2 + y^2$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.0185185	-.0200617	-.0293210	-.0370370	26.9
10	-.0222856	-.0304095	-.0321873	-.0415062	11.4
16	-.0229319	-.0321670	-.0327394	-.0423817	8.8
22	-.0231333	-.0326936	-.0329224	-.0426666	7.9
28	-.0232130	-.0329065	-.0330028	-.0427894	7.5
34	-.0232607	-.0330099	-.0330447	-.0428522	7.4

Table 19

Finite Difference Approximations for $g(x,y) = x^2 + y^2 + x$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.0391738	-.0437441	-.0512227	-.0669516	23.4
10	-.0455917	-.0621758	-.0556990	-.0744159	9.6
16	-.0466739	-.0652647	-.0565486	-.0758624	7.2
22	-.0470070	-.0661833	-.0568259	-.0763281	6.4
28	-.0471457	-.0665529	-.0569463	-.0765270	6.1
34	-.0472151	-.0667312	-.0570084	-.0766281	6.0

Table 20

Galerkin Approximations for $g(x,y) = 10$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.6076859	-.6163949	-.6107867	-.6301380	2.1
10	-.6123007	-.6074556	-.6272431	-.6072230	1.7
16	-.6049013	-.6045481	-.6156474	-.6135822	1.0
22	-.6098773	-.6021091	-.6093475	-.6135225	1.0
28	-.6049922	-.6015982	-.6120686	-.6115899	0.8
34	-.6047867	-.6014126	-.6131791	-.6112015	0.9

Table 21

Galerkin Approximations for $g(x,y) = x^2$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.0123723	-.0231150	-.0125778	-.0239163	1.5
10	-.0126388	-.0235861	-.0126556	-.0230530	0.7
16	-.0128219	-.0232945	-.0125344	-.0231736	1.0
22	-.0129926	-.0232655	-.0124429	-.0232417	1.5
28	-.0128474	-.0232133	-.0124803	-.0231501	1.0
34	-.0128156	-.0232270	-.0124837	-.0231587	1.0

Table 22

Galerkin Approximations for $g(x,y) = x^2 + y^2$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.0293586	-.0363258	-.0318393	-.0473203	7.9
10	-.0281930	-.0354759	-.0344257	-.0476206	4.7
16	-.0269155	-.0353635	-.0349360	-.0473580	3.1
22	-.0271406	-.0357392	-.0350872	-.0474593	3.1
28	-.0262410	-.0357358	-.0356338	-.0476101	1.9
34	-.0262175	-.0356544	-.0356701	-.0475470	1.3

Table 23

Galerkin Approximations for $g(x,y) = x^2 + y^2 + x$

Number of Interior Nodes	x = 1/3 y = 1/3	x = 2/3 y = 1/3	x = 1/3 y = 2/3	x = 2/3 y = 2/3	Average Error (%)
4	-.0542012	-.0715187	-.0569433	-.0835685	4.3
10	-.0530981	-.0715547	-.0596482	-.0832595	2.6
16	-.0521139	-.0709353	-.0599984	-.0831042	1.3
22	-.0527126	-.0712927	-.0599824	-.0833467	2.2
28	-.0515650	-.0711891	-.0605779	-.0833215	1.3
34	-.0514836	-.0711403	-.0606128	-.0832810	1.2

Appendix C

Program Listings

This appendix contains the listings of the microsoft basic programs written for this study. The programs are listed by method for the one- and then the two-dimensional case.

FINITE DIFFERENCE ROUTINE - (ONE-DIMENSION)

```
10 REM ************************************************************************
20 REM THIS PROGRAM UTILIZES A ROUTINE THAT WAS TAKEN FROM  ELEMENTARY
30 REM NUMERICAL ANALYSIS, BY CONTE AND DE BOOR, PG 122; AND
40 REM TRANSLATED INTO BASIC.  ALL VALUES ARE IN DOUBLE-PRECISION.
50 REM ************************************************************************
60 PRINT"ENTER N, THE NUMBER OF STEPS";
70 DEFDBL Q,R,H
80 INPUT N
90 LPRINT"FINITE DIFFERENCE ROUTINE USING DIRECT, TRIDIAGONAL APPROACH"
100 LPRINT
110 LPRINT"N= ";N
120 Q=N
130 H=1#/(Q+1#)
140 DIM A#(N),B#(N),C#(N),D#(N)
150 FOR I= 1 TO N
160 R=I
170 A#(I)=-1#*(Q+1#)^2
180 C#(I)=A#(I)
190 D#(I)=2#*(Q+1#)^2
200 B#(I)=1#+4#*(R*H)^2
210 NEXT I
220 GOSUB 290
230 PRINT"THE SOLUTION IS:"
240 FOR I= 1 TO N
250 PRINT I,B#(I)
260 LPRINT I,B#(I)
270 NEXT I
280 END
290 IF N>1 THEN 320
300 B#(1)=B#(1)/D#(1)
310 RETURN
320 FOR I= 2 TO N
330 R=-A#(I)/D#(I-1)
340 D#(I)= D#(I)+R*C#(I-1)
350 B#(I)=B#(I)+R*B#(I-1)
360 NEXT I
370 B#(N)=B#(N)/D#(N)
380 K=N
390 FOR J= 2 TO N
400 K=K-1
410 B#(K)=(B#(K)-C#(K)*B#(K+1))/D#(K)
420 NEXT J
430 RETURN
```

FINITE DIFFERENCE ROUTINE - (TWO-DIMENSION)

```
1 REM*****************************************************************
2 REM THIS PROGRAM USES AN INVERSION ROUTINE TAKEN FROM NUMERICAL
3 REM METHODS BY R.W. HORNBECK, PG 294-295. THE ROUTINE EMPLOYS
4 REM GAUSS-JORDAN ELIMINATION WITH COLUMN SHIFTING TO MAXIMIZE
5 REM PIVOT ELEMENTS
6 REM*****************************************************************
10 DEFDBL S,T,H,K,A-G,X,Y
20 PRINT CHR$(26)+CHR$(27)+CHR$(13)
30 PRINT"TWO-DIMENSIONAL FINITE DIFFERENCE GREEN'S FUNCTION ROUTINE"
40 PRINT:PRINT:PRINT
50 PRINT"INPUT S, THE # OF X DIVISIONS; AND T, THE # OF Y DIVISIONS"
60 INPUT S,T
70 PRINT S,T
80 N=S*T
90 LPRINT"N=";N
100 H=1#/(S+1#):K=1#/(T+1#)
110 PRINT"H=";H;"K=";K
120 DIM G#(N,N),J#(N+25),F#(N),U#(N)
130 REM*****************************************************************
140 REM CREATES INITIAL COEFFICIENT MATRIX FOR THE GREEN'S FUNCTION
150 REM*****************************************************************
160 FOR I=1 TO N
170 IF I MOD S =1 THEN 190
180 G#(I,I-1)=K*K
190 G#(I,I)=-2#*(K*K+H*H)
200 IF I MOD S=0 THEN 220
210 G#(I,I+1)=K*K
220 IF (I-S)<=0 THEN 240
230 G#(I,I-S)=H*H
240 IF (I+S)>=N THEN 260
250 G#(I,I+S)=H*H
260 NEXT I
270 FOR I=1 TO N
280 FOR J=1 TO N
290 PRINT G#(I,J);"    ";
300 NEXT J
310 PRINT
320 NEXT I
330 REM*************************
340 REM MATRIX INVERSION ROUTINE
350 REM*************************
360 PD=1
370 FOR L=1 TO N
380 D=0
390 FOR P=1 TO N
400 D=D+G#(L,P)*G#(L,P)
410 NEXT P
420 D=SQR(D)
430 PD=PD*D
440 NEXT L
450 DETM=1
460 FOR L=1 TO N
```

```
470 Q=L
480 J#(L+20)=Q
490 NEXT L
500 FOR L=1 TO N
510 C=0:M=L
520 FOR P=L TO N
530 IF (ABS(C)-ABS(G#(L,P)))>=0 THEN 560
540 M=P
550 C=G#(L,P)
560 NEXT P
570 IF L=M THEN 660
580 R=J#(M+20)
590 J#(M+20)=J#(L+20)
600 J#(L+20)=R
610 FOR P=1 TO N
620 S=G#(P,L)
630 G#(P,L)=G#(P,M)
640 G#(P,M)=S
650 NEXT P
660 G#(L,L)=1#
670 DETM=DETM*C
680 FOR M=1 TO N
690 G#(L,M)=G#(L,M)/C
700 NEXT M
710 FOR M=1 TO N
720 IF L=M THEN 790
730 C=G#(M,L)
740 IF C=0 THEN 790
750 G#(M,L)=0
760 FOR P=1 TO N
770 G#(M,P)=G#(M,P)-C*G#(L,P)
780 NEXT P
790 NEXT M
800 NEXT L
810 FOR L=1 TO N
820 Q=L
830 IF J#(L+20)=Q THEN 950
840 M=L
850 M=M+1
860 IF J#(M+20)=Q THEN 880
870 IF N>M THEN 850
880 J#(M+20)=J#(L+20)
890 FOR P=1 TO N
900 C=G#(L,P)
910 G#(L,P)=G#(M,P)
920 G#(M,P)=C
930 NEXT P
940 J#(L+20)=Q
950 NEXT L
960 DETM=ABS(DETM)
970 DTNRM=DETM/PD
980 PRINT"DTNRM=";DTNRM
```

```
1000 REM*************************************************************
1010 REM  MULTIPLIES INVERTED MATRIX BY H*K*IDENTITY MATRIX
1020 REM*************************************************************
1030 FOR I=1 TO N
1040 FOR J=1 TO N
1050 G#(I,J)=G#(I,J)*H*K
1060 NEXT J
1070 NEXT I
1180 PRINT:PRINT:PRINT"COLUMN VECTOR OF EQUATION"
1190 REM*************************************************************************
1200 REM CREATES COLUMN VECTOR OF EQUATION TO BE EVALUATED, MULTIPLIED BY h*k
1210 REM*************************************************************************
1220 PRINT"THE FUNCTIONS TO BE EVALUATED ARE OF THE FORM: AX^2+BY^2+CX+DY+E"
1230 PRINT"INPUT A":INPUT A
1240 PRINT"INPUT B":INPUT B
1250 PRINT"INPUT C":INPUT C
1260 PRINT"INPUT D":INPUT D
1270 PRINT"INPUT E":INPUT E
1280 FOR I=1 TO N
1290 Q=I
1300 IF I<=N/2 THEN Y=1/(T+1) ELSE Y=2/(T+1)
1310 IF I<=N/2 THEN X=Q/(S+1) ELSE X=(Q-N/2)/(S+1)
1320 F#(I)=(A*X^2+B*Y^2+C*X+D*Y+E)*H*K
1330 PRINT F#(I)
1340 NEXT I
1350 PRINT:PRINT:PRINT"SOLUTION MATRIX"
1360 FOR I=1 TO N
1370 U#(I)=0
1380 FOR J=1 TO N
1390 U#(I)=U#(I)+G#(I,J)*F#(J)
1400 NEXT J
1410 LPRINT U#(I)
1420 NEXT I
1430 END
```

GALERKIN ROUTINE - (ONE-DIMENSION)

```
10 REM ************************************************************************
20 REM THIS PROGRAM USES A DIRECT GAUSSIAN ELIMINATION ROUTINE WITH
30 REM PIVOTING TAKEN FROM APPLIED NUMERICAL METHODS FOR DIGITAL COMPUTATION
40 REM BY JAMES, SMITH, & WOLFORD, PG 192-193. IN ADDITION, THE INITIAL
50 REM COEFFICIENT MATRIX HAS BEEN MODIFIED SO THAT THE LARGER NUMBERS ARE
60 REM IN THE UPPER LEFT OF THE MATRIX; THIS WAS DONE TO MAXIMIZE THE
70 REM EFFECTIVENESS OF THE PIVOTING.
80 REM ************************************************************************
90 PRINT CHR$(26)+CHR$(27)+CHR$(13)
100 PRINT"GALERKIN (METHOD OF MOMENTS) ROUTINE"
110 PRINT:PRINT:PRINT
120 PRINT"INPUT N, THE NUMBER OF DIVISIONS";
130 INPUT N
140 LPRINT"N= ";N
150 M=N+1
160 L=N-1
170 DIM A#(N,M),X#(N),ALPHA#(N),U#(N)
180 DEFDBL Q-T,Y
190 REM*************************************************************************
200 REM******THIS CALCULATES THE INITIAL L MATRIX*************
210 FOR I=1 TO N
220 FOR J=1 TO N
230 Q=M-I:R=M-J
240 A#(I,J)=(Q*R)/(Q+R+1#)
250 NEXT J
260 NEXT I
270 PRINT:PRINT:PRINT"PICK A FUNCTION TO EVALUATE"
280 PRINT:PRINT"1- G=10"
290 PRINT"2- G=X^2"
300 PRINT"3- G=X^2 + 1"
310 PRINT"4- G=X^2 + X + 1"
320 PRINT"5- G=4X^2 + 1"
330 INPUT Z
340 FOR I=1 TO N
350 S=M-I
360 ON Z GOSUB 960,980,1000,1020,1040
370 NEXT I
380 FOR K=1 TO L
390 H=K
400 R=ABS(A#(K,K))
410 B=K+1
420 FOR I=B TO N
430 S=ABS(A#(I,K))
440 IF (R-S)>=0 THEN 460
450 R=S:H=I
460 NEXT I
470 IF (H-K)=0 THEN 530
480 FOR J=K TO M
490 Q=A#(H,J)
500 A#(H,J)=A#(K,J)
510 A#(K,J)=Q
520 NEXT J
```

```
530 FOR I=B TO N
540 Q=A#(I,K)/A#(K,K)
550 FOR J=B TO M
560 A#(I,J)=A#(I,J)-Q*A#(K,J)
570 NEXT J
580 NEXT I
590 FOR I=B TO N
600 A#(I,K)=0
610 NEXT I
620 NEXT K
630 X#(N)=A#(N,M)/A#(N,N)
640 FOR O=1 TO L
650 T=0
660 I=N-O
670 C=I+1
680 FOR J=C TO N
690 T=T+A#(I,J)*X#(J)
700 NEXT J
710 X#(I)=(A#(I,M)-T)/A#(I,I)
720 NEXT O
730 FOR I=1 TO N
740 ALPHA#(I)=X#(M-I)
750 NEXT I
760 PRINT:PRINT:PRINT"INPUT THE POINT YOU WANT EVALUATED";
770 INPUT Y
780 X=Y
790 FOR I=1 TO N
800 U#(I)=Y-Y^(I+1)
810 NEXT I
820 S=0
830 FOR I=1 TO N
840 Q=U#(I)*ALPHA#(I)
850 S=S+Q
860 NEXT I
870 LPRINT"U(";X;")= ";S
880 PRINT:PRINT"DO YOU WANT TO EVALUATE ANOTHER POINT";
890 INPUT Y$
900 IF Y$="Y" THEN 760
910 END
920 REM*************************************************************************
930 REM THESE SUBROUTINES CALCULATE THE INHOM. TERM TO ADD TO THE AUGMENTED
940 REM COEFFICIENT MATRIX
950 REM*************************************************************************
960 A#(I,N+1)=5#*S/(S+2#)
970 RETURN
980 A#(I,N+1)=S/(4#*(S+4#))
990 RETURN
1000 A#(I,N+1)=S*(3#*S+10#)/(4#*(S+4#)*(S+2#))
1010 RETURN
1020 A#(I,N+1)=(13#*(S+4#)*(S+3#)*(S+2#)-12#*(3#*S^2+18#*S+26#))/(12#*(S+4#)
     *(S+3#)*(S+2#))
1030 RETURN
1040 A#(I,N+1)=S*(3#*S+8#)/(2#*(S+2#)*(S+4#))
1050 RETURN
```

GALERKIN ROUTINE - (TWO-DIMENSION)

```
10 REM ************************************************************************
20 REM THIS PROGRAM USES A DIRECT GAUSSIAN ELIMINATION ROUTINE WITH
30 REM PIVOTING TAKEN FROM APPLIED NUMERICAL METHODS FOR DIGITAL COMPUTATION
40 REM BY JAMES, SMITH, & WOLFORD, PG 192-193.
50 REM ************************************************************************
60 PRINT CHR$(26)+CHR$(27)+CHR$(13)
70 PRINT"GALERKIN (METHOD OF MOMENTS) ROUTINE"
80 PRINT:PRINT:PRINT
90 PRINT"INPUT N, THE NUMBER OF DIVISIONS";
100 INPUT N
110 LPRINT"N= ";N
120 M=N+1
130 L=N-1
140 DIM A#(N,M),X#(N),ALPHA#(N),U#(N)
150 DEFDBL A-H,Q-T,Y,X
160 REM*************************************************************************
170 REM******THIS CALCULATES THE INITIAL L MATRIX*************
180 FOR I=1 TO N
190 FOR J=1 TO N
200 Q=I:R=J:S=2*(Q+R)
210 P=I MOD 3:V=J MOD 3
220 IF (P=1 AND V=1) THEN GOSUB 1200
230 IF (P=1 AND V=2) THEN GOSUB 1250
240 IF (P=2 AND V=1) THEN 250 ELSE 270
250 Q=J:R=I
260 GOSUB 1250
270 IF (P=1 AND V=0) THEN GOSUB 1320
280 IF (P=0 AND V=1) THEN 290 ELSE 310
290 Q=J:R=I
300 GOSUB 1320
310 IF (P=2 AND V=2) THEN GOSUB 1390
320 IF (P=2 AND V=0) THEN GOSUB 1440
330 IF (P=0 AND V=2) THEN 340 ELSE 360
340 Q=J:R=I
350 GOSUB 1440
360 IF (P=0 AND V=0) THEN GOSUB 1510
370 NEXT J
380 NEXT I
390 PRINT CHR$(7)
400 PRINT:PRINT"THE FUNCTIONS TO EVALUATE ARE OF THE FORM: AX^2+BY^2+CX+DY+E"
410 PRINT"INPUT A";
420 INPUT A
430 PRINT"INPUT B";
440 INPUT B
450 PRINT"INPUT C";
460 INPUT C
470 PRINT"INPUT D";
480 INPUT D
490 PRINT"INPUT E";
500 INPUT E
510 FOR I=1 TO N
520 Q=I:R=2*Q
```

```
530 P=Q MOD 3
540 IF P=1 THEN GOSUB 1560
550 IF P=2 THEN GOSUB 1620
560 IF P=0 THEN GOSUB 1680
570 NEXT I
580 FOR K=1 TO L
590 H=K
600 R=ABS(A#(K,K))
610 B=K+1
620 FOR I=B TO N
630 S=ABS(A#(I,K))
640 IF (R-S)>=0 THEN 660
650 R=S:H=I
660 NEXT I
670 IF (H-K)=0 THEN 730
680 FOR J=K TO M
690 Q=A#(H,J)
700 A#(H,J)=A#(K,J)
710 A#(K,J)=Q
720 NEXT J
730 FOR I=B TO N
740 Q=A#(I,K)/A#(K,K)
750 FOR J=B TO M
760 A#(I,J)=A#(I,J)-Q*A#(K,J)
770 NEXT J
780 NEXT I
790 FOR I=B TO N
800 A#(I,K)=0
810 NEXT I
820 NEXT K
830 X#(N)=A#(N,M)/A#(N,N)
840 FOR O=1 TO L
850 T=0
860 I=N-O
870 C=I+1
880 FOR J=C TO N
890 T=T+A#(I,J)*X#(J)
900 NEXT J
910 X#(I)=(A#(I,M)-T)/A#(I,I)
920 NEXT O
930 FOR I=1 TO N
940 ALPHA#(I)=X#(I)
950 NEXT I
960 PRINT CHR$(7)
970 PRINT:PRINT:PRINT"INPUT THE X,Y POINTS YOU WANT EVALUATED";
980 INPUT X,Y
990 FOR I=1 TO N
1000 Q=I
1010 P=I MOD 3
1020 IF P=1 THEN U#(I)=(X*Y)^((2#*Q+1#)/3#)*(1#-X)*(1#-Y)
1030 IF P=2 THEN U#(I)=X^((2#*Q+5#)/3#)*Y*(1#-X)*(1#-Y)
1040 IF P=0 THEN U#(I)=Y^((2#*Q+3#)/3#)*X*(1#-X)*(1#-Y)
1050 NEXT I
1060 S=0
```

```
1070 FOR I=1 TO N
1080 Q=U#(I)*ALPHA#(I)
1090 S=S+Q
1100 NEXT I
1110 LPRINT "U(";X;Y;")=";S
1120 PRINT:PRINT"DO YOU WANT TO EVALUATE ANOTHER POINT";
1130 INPUT Y$
1140 IF Y$="Y" THEN 970
1150 END
1160 REM*************************************************************************
1170 REM THE FOLLOWING SUBROUTINES ARE USED TO CREATE THE AUGMENTED LMN
1180 REM COEFFICIENT MATRIX
1190 REM*************************************************************************
1200 REM CALCULATES LMN MATRIX VALUES FOR M=1,4,7,... AND N=1,4,7,...
1210 A=1#/(S+5#)-2#/(S+8#)+1#/(S+11#)
1220 B=(R-1#)/(S-1#)-(2#*R+1#)/(S+2#)+(R+2#)/(S+5#)
1230 A#(I,J)=4#*(2#*R+1#)*A*B
1240 RETURN
1250 REM CALCULATES LMN MATRIX VALUES FOR M=1,4,7,... AND N=2,5,8,...
1260 A=1#/(2#*Q+7#)-2#/(2#*Q+10#)+1#/(2#*Q+13#)
1270 B=(R+1#)/(S+3#)-(2#*R+5#)/(S+6#)+(R+4#)/(S+9#)
1280 C=1#/(S+9#)-2#/(S+12#)+1#/(S+15#)
1290 D=1#/(2#*Q+7#)-1#/(2#*Q+4#)
1300 A#(I,J)=(2#*A*B*(2#*R+5#)+18#*C*D)
1310 RETURN
1320 REM CALCULATES LMN MATRIX VALUES FOR M=1,4,7,... AND N=3,6,9,...
1330 A=1#/(2#*Q+7#)-2#/(2#*Q+10#)+1#/(2#*Q+13#)
1340 B=R/(S+1#)-(2#*R+3#)/(S+4#)+(R+3#)/(S+7#)
1350 C=1#/(S+7#)-2#/(S+10#)+1#/(S+13#)
1360 D=1#/(2#*Q+7#)-1#/(2#*Q+4#)
1370 A#(I,J)=(2#*A*B*(2#*R+3#)+18#*C*D)
1380 RETURN
1390 REM CALCULATES LMN MATRIX VALUES FOR M=2,5,8,... AND N=2,5,8,...
1400 A=(R+1#)/(S+7#)-(2#*R+5#)/(S+10#)+(R+4#)/(S+13#)
1410 B=1#/(S+13#)-2#/(S+16#)+1#/(S+19#)
1420 A#(I,J)=(A*(2#*R+5#)/45#-B)
1430 RETURN
1440 REM CALCULATES LMN MATRIX VALUES FOR M=2,5,8,... AND N=3,6,9,...
1450 A=1#/(2#*Q+11#)-2#/(2#*Q+14#)+1#/(2#*Q+17#)
1460 B=R/(2#*R+3#)-(2#*R+3#)/(2#*R+6#)+(R+3#)/(2#*R+9#)
1470 C=1#/(2#*R+9#)-2#/(2#*R+12#)+1#/(2#*R+15#)
1480 D=1#/(2#*Q+11#)-1#/(2#*Q+8#)
1490 A#(I,J)=(2#*A*B*(2#*R+3#)+18#*C*D)
1500 RETURN
1510 REM CALCULATES LMN MATRIX VALUES FOR M=3,6,9,... AND N=3,6,9,...
1520 A=R/(S+3#)-(2#*R+3#)/(S+6#)+(R+6#)/(S+9#)
1530 B=1#/(S+9#)-2#/(S+12#)+1#/(S+15#)
1540 A#(I,J)=(A*(2#*R+3#)/45#-B)
1550 RETURN
1560 REM CALCULATES GM FOR M=1,4,7,...
1570 F=(A+B)*(1#/(R+10#)-1#/(R+13#))
1580 G=(C+D)*(1#/(R+7#)-1#/(R+10#))
1590 H=E*(1#/((R+4#)*(R+4#))-2#/((R+4#)*(R+7#))+1#/((R+7#)*(R+7#)))
1600 A#(I,N+1)=9#*((F+G)*(1#/(R+4#)-1#/(R+7#))+H)
```

```
1610 RETURN
1620 REM CALCULATES GM FOR M=2,5,8,...
1630 F=A/2#*(1#/(R+14#)-1#/(R+17#))
1640 G=C/2#*(1#/(R+11#)-1#/(R+14#))
1650 H=(1#/(R+8#)-1#/(R+11#))*(3#*B/20#+D/4#+E/2#)
1660 A#(I,N+1)=F+G+H
1670 RETURN
1680 REM CALCULATES GM FOR M=3,6,9,...
1690 F=(1#/(R+6#)-1#/(R+9#))*(3#*A/20#+C/4#+E/2#)
1700 G=B/2#*(1#/(R+12#)-1#/(R+15#))
1710 H=D/2#*(1#/(R+9#)-1#/(R+12#))
1720 A#(I,N+1)=F+G+H
1730 RETURN
```

COLLOCATION ROUTINE - (ONE-DIMENSION)

```
10 REM ************************************************************************
20 REM THIS PROGRAM USES AN INVERSION ROUTINE TAKEN FROM NUMERICAL METHODS
30 REM BY R.W. HORNBECK, PG 294-295. THE ROUTINE EMPLOYS GAUSS-JORDAN
40 REM ELIMINATION WITH COLUMN SHIFTING TO MAXIMIZE PIVOT ELEMENTS.
50 REM ************************************************************************
60 PRINT CHR$(26)+CHR$(27)+CHR$(13)
70 LPRINT"CO-LOCATION (METHOD OF MOMENTS) ROUTINE"
80 PRINT:PRINT:LPRINT
90 PRINT"INPUT N, THE NUMBER OF DIVISIONS";
100 INPUT N
110 LPRINT"N= ";N
120 DIM C#(N,N),ALPHA#(N),G#(N),J#(N+25),U#(N)
130 DEFDBL Q-S,Y,C,D
140 REM************************************************************************
150 REM C#(N,N) CONTAINS THE INITIAL L MATRIX, THEN THE INVERTED L MATRIX;
160 REM J#(N+25) IS USED ONLY IN THE INVERSION ROUTINE
170 REM************************************************************************
180 REM******THIS CALCULATES THE INITIAL L MATRIX*************
190 FOR I=1 TO N
200 FOR J=1 TO N
210 Q=I:R=J:S=N
220 C#(I,J)=R*(R+1#)*(Q/(S+1#))^(R-1#)
230 NEXT J
240 NEXT I
250 PD=1
260 FOR L=1 TO N
270 D=0
280 FOR K=1 TO N
290 D=D+C#(L,K)*C#(L,K)
300 NEXT K
310 D=SQR(D)
320 PD=PD*D
330 NEXT L
340 DETM=1
350 FOR L=1 TO N
360 Q=L
370 J#(L+20)=Q
380 NEXT L
390 FOR L=1 TO N
400 C=0
410 M=L
420 FOR K=L TO N
430 IF (ABS(C)-ABS(C#(L,K))) >= 0 THEN 460
440 M=K
450 C=C#(L,K)
460 NEXT K
470 IF L=M THEN 560
480 R=J#(M+20)
490 J#(M+20)=J#(L+20)
500 J#(L+20)=R
510 FOR K=1 TO N
520 S=C#(K,L)
```

```
530 C#(K,L)=C#(K,M)
540 C#(K,M)=S
550 NEXT K
560 C#(L,L)=1#
570 DETM=DETM*C
580 FOR M=1 TO N
590 C#(L,M)=C#(L,M)/C
600 NEXT M
610 FOR M=1 TO N
620 IF L=M THEN 690
630 C=C#(M,L)
640 IF C=0# THEN 690
650 C#(M,L)=0
660 FOR K=1 TO N
670 C#(M,K)=C#(M,K)-C*C#(L,K)
680 NEXT K
690 NEXT M
700 NEXT L
710 FOR L=1 TO N
720 Q=L
730 IF J#(L+20)=Q THEN 850
740 M=L
750 M=M+1
760 IF J#(M+20)=Q THEN 780
770 IF N>M THEN 750
780 J#(M+20)=J#(L+20)
790 FOR K=1 TO N
800 C=C#(L,K)
810 C#(L,K)=C#(M,K)
820 C#(M,K)=C
830 NEXT K
840 J#(L+20)=Q
850 NEXT L
860 DETM=ABS(DETM)
870 DTNRM=DETM/PD
880 REM*********CALCULATES gm, THE INNER PRODUCT OF THE WEIGHT FN & g*******
890 FOR I=1 TO N
900 Q=I:R=N
910 REM******THE NEXT LINE CAN BE CHANGED FOR OTHER INHOMOGENEITIES******
920 G#(I)=1#+4#*(Q/(R+1#))^2
930 PRINT"G(";I;")= ";G#(I)
940 NEXT I
950 REM*******CALCULATES ALPHA********
960 FOR I=1 TO N
970 ALPHA#(I)=0
980 FOR J=1 TO N
990 ALPHA#(I)=ALPHA#(I)+C#(I,J)*G#(J)
1000 NEXT J
1010 PRINT"ALPHA(";I;")= ";ALPHA#(I)
1020 NEXT I
1030 PRINT"INPUT THE THE POINT YOU WANT EVALUATED";
1040 INPUT Y
1050 X=Y
1060 FOR I=1 TO N
```

```
1070 U#(I)=Y-Y^(I+1)
1080 PRINT U#(I)
1090 NEXT I
1100 S=0
1110 FOR I=1 TO N
1120 Q=U#(I)*ALPHA#(I)
1130 S=S+U#(I)*ALPHA#(I)
1140 PRINT"U(";I;")= ";Q
1150 NEXT I
1160 PRINT"U(";X;")= ";S
1170 LPRINT"U(";X;")= ";S
1180 PRINT:PRINT"DO YOU WANT TO EVALUATE ANOTHER POINT (Y,N)";
1190 INPUT Y$
1200 IF Y$="Y" THEN 1030
1210 END
```

COLLOCATION ROUTINE - (TWO-DIMENSION)

```
10 REM ************************************************************************
20 REM THIS PROGRAM USES A DIRECT GAUSSIAN ELIMINATION ROUTINE WITH
30 REM PIVOTING TAKEN FROM APPLIED NUMERICAL METHODS FOR DIGITAL
40 REM COMPUTATION BY JAMES, SMITH & WOLFORD, PG 192-193.
50 REM ************************************************************************
60 PRINT CHR$(26)+CHR$(27)+CHR$(13)
70 PRINT:PRINT:PRINT
80 DEFDBL T,Z
90 PRINT"INPUT THE NUMBER OF X DIVISIONS, AND THE NUMBER OF Y DIVISIONS"
100 INPUT T,Z
110 N=Z*T
120 M=N+1:L=N-1
130 LPRINT"N=";N
140 DIM C#(N,M),ALPHA#(N),X#(N),U#(N)
150 DEFDBL Q-S,X,Y,A-E,V,T
160 REM*************************************************************************
170 REM******THIS CALCULATES THE INITIAL L MATRIX************ ***********
180 REM*************************************************************************
210 FOR I=1 TO N
220 V=I
230 IF I<=N/2 THEN Y=1/(Z+1) ELSE Y=2/(Z+1)
250 IF I<=N/2 THEN X=V/(T+1) ELSE X=(V-N/2)/(T+1)
270 FOR J=1 TO N
280 Q=J
290 P=J MOD 3
300 IF P=1 THEN GOSUB 1140
310 IF P=2 THEN GOSUB 1210
320 IF P=0 THEN GOSUB 1260
340 NEXT J
360 NEXT I
370 REM*********CALCULATES gm, THE INNER PRODUCT OF THE WEIGHT FN & g*******
375 PRINT CHR$(7)
380 PRINT:PRINT"THE FUNCTIONS TO BE EVALUATED ARE OF THE FORM:  AX^2 + BY^2
    + CX + DY +E"
390 PRINT"INPUT A";
400 INPUT A
410 PRINT"INPUT B";
420 INPUT B
430 PRINT"INPUT C";
440 INPUT C
450 PRINT"INPUT D";
460 INPUT D
470 PRINT"INPUT E";
480 INPUT E
490 FOR I=1 TO N
500 Q=I
510 IF I<=N/2 THEN Y=1/(Z+1) ELSE Y=2/(Z+1)
520 IF I<=N/2 THEN X=Q/(T+1) ELSE X=(Q-N/2)/(T+1)
530 C#(I,M)=A*X*X + B*Y*Y + C*X + D*Y + E
540 NEXT I
550 FOR K=1 TO L
560 H=K
```

```
570 R=ABS(C#(K,K))
580 B=K+1
590 FOR I=B TO N
600 S=ABS(C#(I,K))
610 IF (R-S)>=0 THEN 630
620 R=S:H=I
630 NEXT I
640 IF (H-K)=0 THEN 700
650 FOR J=K TO M
660 Q=C#(H,J)
670 C#(H,J)=C#(K,J)
680 C#(K,J)=Q
690 NEXT J
700 FOR I=B TO N
710 Q=C#(I,K)/C#(K,K)
720 FOR J=B TO M
730 C#(I,J)=C#(I,J)-Q*C#(K,J)
740 NEXT J
750 NEXT I
760 FOR I=B TO N
770 C#(I,K)=0
780 NEXT I
790 NEXT K
800 X#(N)=C#(N,M)/C#(N,N)
810 FOR O=1 TO L
820 S=0
830 I=N-O
840 C=I+1
850 FOR J=C TO N
860 S=S+C#(I,J)*X#(J)
870 NEXT J
880 X#(I)=(C#(I,M)-S)/C#(I,I)
890 NEXT O
900 FOR I=1 TO N
910 ALPHA#(I)=X#(I)
920 NEXT I
925 PRINT CHR$(7)
930 PRINT"INPUT THE X,Y POINT YOU WANT EVALUATED";
940 INPUT X,Y
950 FOR I=1 TO N
960 Q=I
970 P=I MOD 3
980 IF P=1 THEN U#(I)=(X*Y)^((2#*Q+1#)/3#)*(1#-X)*(1#-Y)
990 IF P=2 THEN U#(I)=X^((2#*Q+5#)/3#)*Y*(1#-X)*(1#-Y)
1000 IF P=0 THEN U#(I)=Y^((2#*Q+3#)/3#)*X*(1#-X)*(1#-Y)
1010 NEXT I
1020 S=0
1030 FOR I=1 TO N
1040 Q=U#(I)*ALPHA#(I)
1050 S=S+U#(I)*ALPHA#(I)
1060 PRINT"U(";I;")= ";Q
1070 NEXT I
1080 PRINT"U(";X;Y;")= ";S
1090 LPRINT"U(";X;Y;")= ";S
```

```
1100 PRINT:PRINT"DO YOU WANT TO EVALUATE ANOTHER POINT (Y,N)";
1110 INPUT Y$
1120 IF Y$="Y" THEN 930
1130 END
1140 REM CALCULATES THE LMN MATRIX VALUES FOR N=1,4,7,...
1150 A=X^((2#*Q-5#)/3#)*(Y^((2#*Q+1#)/3#)-Y^((2#*Q+4#)/3#))
1160 B=Y^((2#*Q-5#)/3#)*(X^((2#*Q+1#)/3#)-X^((2#*Q+4#)/3#))
1170 C=X^((2#*Q-2#)/3#)*(Y^((2#*Q+4#)/3#)-Y^((2#*Q+1#)/3#))
1180 D=Y^((2#*Q-2#)/3#)*(X^((2#*Q+4#)/3#)-X^((2#*Q+1#)/3#))
1190 C#(I,J)=((2#*Q-2#)*(A+B)+(2#*Q+4#)*(C+D))*(2#*Q+1#)/9#
1200 RETURN
1210 REM CALCULATES THE LMN MATRIX VALUES FOR N=2,5,8,...
1220 A=(2#*Q+2#)*X^((2#*Q-1#)/3#)-(2#*Q+8#)*X^((2#*Q+2#)/3#)
1230 B=2#*(X^((2#*Q+8#)/3#)-X^((2#*Q+5#)/3#))
1240 C#(I,J)=A*(2#*Q+5#)*(Y-Y*Y)/9#+B
1250 RETURN
1260 REM CALCULATES THE LMN MATRIX VALUES FOR N=3,6,9,...
1270 A=(2#*Q)*Y^((2#*Q-3#)/3#)-(2#*Q+6#)*Y^(2#*Q/3#)
1280 B=2#*(Y^((2#*Q+6#)/3#)-Y^((2#*Q+3#)/3#))
1290 C#(I,J)=A*(2#*Q+3#)*(X-X*X)/9#+B
1300 RETURN
```

VITA

Dean E. Oyler was born on 3 October 1956 in Blythesville, Arkansas. He attended the University of Central Florida from which he received a Bachelor of Science degree in Physics, in June 1981. Upon graduation, he received a commission in the USAF through the ROTC program and entered active duty in October 1981. His first assignment was to the Aeronautical Systems Division at Wright-Patterson AFB, OH. He worked as an Operations Research Analyst in the Research and Cost Division (ASD/ACCR) until being reassigned to the Air Force Institute of Technology, WPAFB, OH in May 1983.

Permanent address: 1615 Idaho Avenue
Orlando, FL 32809

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GEP/PH/84D-7	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENP	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification): See Box 19	
12. PERSONAL AUTHOR(S): Dean E. Oyler, B.A., B.S., 1/Lt, USAF	
13a. TYPE OF REPORT: MS Thesis; 13b. TIME COVERED: FROM _____ TO _____; 14. DATE OF REPORT (Yr., Mo., Day): 1984 December; 15. PAGE COUNT: 81	
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Green's Functions, Method of Weighted Residuals Method of Galerkin, Method of Collocation, Method of Finite Differences
12	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: INVESTIGATION OF THE NUMERICAL METHOD OF MOMENTS FOR DIGITAL COMPUTER DETERMINATION OF DIFFERENTIAL EQUATIONS

Approved for public release: IAW AFR 190-17.
21 Feb 85
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Dr. Bernard Kaplan	22b. TELEPHONE NUMBER (Include Area Code): 513-255-4498; 22c. OFFICE SYMBOL: AFIT/ENP

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

The purpose of this study was to determine the feasibility of using the method of weighted residuals to obtain approximations to the discrete Green's function, or analogs to it. The weighted residual methods of Galerkin and collocation, ad well as the finite difference method were programed on a Kaypro II micro-computer in Microsoft Basic. These programs were used to generate approximations to the one- and two-dimensional Poisson's equation. The two-dimensional case was restricted to the geometry of a unit square. Various inhomogeneity terms were used to obtain approximate solutions to the discrete Green's functions or their analogs. The results were compared with the analytical values at various interior nodal points on the mesh. The average percent error for the approximations were reported for each case as the number of interior nodal points was increased. The areas of consideration were the rate of convergence to the analytical solution, the amount of time it took to run each program, and the accuracy of the approximate solutions. The results of this study indicate that the Green's functions or analogs obtained are valid approximations to the discrete Green's functions. The method of weighted residuals proved to be very sensitive to the choice of basis functions, resulting in ill-conditioned matricies in some instances.

END

FILMED

5-85

DTIC